

基于随机模型的服务质量 多属性评价与优化

黄霁崑 著

電子工業出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

随着服务计算技术的蓬勃发展,服务质量越发成为重要的需求,并呈现多样化特性。本书以理论模型、量化分析和优化技术为重点,从性能、可信赖性和能耗三个维度开展服务质量的多属性评价和优化研究。本书以随机模型为主线,从服务内部的特性出发,描述服务的动态行为,通过数学建模和分析方法,建立服务质量多维度属性的量化评价,进一步分析讨论模型参数和评价结果之间的量化关系,并通过优化方法和技术得到最佳的系统参数或配置方案。在服务质量评价方面,本书介绍了基于排队模型的性能指标评价方法、基于半马尔可夫模型的可信赖性评价方法和基于随机游走模型的服务质量排序方法;在服务质量优化方面,本书介绍了基于折扣马尔可夫决策的性能-能耗综合优化方法,以及基于平均时间马尔可夫决策的可信赖性-能耗综合优化方法。

本书旨在为系统性能评价和服务计算领域的研究人员提供专业参考。本书可作为高等学校相关专业研究生的教学用书,也可作为相关领域研究人员的理论和技术指导书。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有,侵权必究。

图书在版编目(CIP)数据

基于随机模型的服务质量多属性评价与优化 / 黄霁巍著. —北京: 电子工业出版社, 2017. 8
ISBN 978-7-121-32523-6

I. ①基… II. ①黄… III. ①互联网络-网络服务器 IV. ①TP368.5

中国版本图书馆 CIP 数据核字 (2017) 第 203392 号

策划编辑: 张 剑 (zhang@phei.com.cn)

责任编辑: 张 剑

印 刷:

装 订:

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本: 787 × 1092 1/16 印张: 6.75 字数: 173 千字

版 次: 2017 年 8 月第 1 版

印 次: 2017 年 8 月第 1 次印刷

定 价: 68.00 元

凡所购买电子工业出版社图书有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系, 联系及邮购电话: (010)88254888, 88258888。

质量投诉请发邮件至 zltz@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

本书咨询联系方式: zhang@phei.com.cn。

前 言

随着计算机技术和网络技术的不断发展，服务计算（Services Computing）应运而生。服务计算是一种新兴的计算模式，它使服务提供商将特定的功能封装成为遵循统一标准的、交互式的计算机程序，将其以服务的形式提供给不同商业领域中的多个用户使用。服务计算具有灵活、高效的特点，构成了连接商业服务与信息服务的桥梁，带来了商业模式和计算模式的转型，受到了学术界和工业界的青睐。随着服务计算技术的发展和广泛应用，互联网上各类服务的数目逐渐增多。越来越多的第三方在互联网上提供了功能相同或相似的服务，为同样的服务流程提供了多个候选服务。因此，用户对服务的关注逐渐从功能性需求（服务标准的制定、服务流程的设计、服务系统的实现等）向非功能性需求（服务质量）转变。服务质量（Quality of Service, QoS）体现的是消费者对服务提供者所提供的服务的满意程度，是对服务者服务水平的一种度量和评价。目前，在服务计算中按照用户的要求提供 QoS 保障是一个普遍的需求，也是服务计算发展的重要挑战，已成为当今服务计算领域的核心研究方向和热点问题之一。

随着服务计算应用范围的扩展，用户对服务质量的需求逐渐呈现多样化的特点。首先，性能是服务质量首要关注的指标，它反映了服务的效率，代表服务的处理能力，对于计算密集型服务或实时服务具有重要意义，是其核心的服务质量评价指标。服务质量在性能维度上又包括响应时间、吞吐率、利用率等属性。其次，随着服务计算逐渐应用于医疗卫生、交通管理或军事作战等领域，用户对服务正常持续运行的能力提出了越来越高的要求，因此可信性作为其评价指标成为服务质量中的另一类重要需求。可信性代表了服务系统避免或减少发生服务失效的能力，包括可用性、完整性、保险性、可维护性和可靠性等属性。此外，随着支撑服务的计算机和网络系统的规模逐渐增大，能源消耗越来越大，对环境造成的影响越来越严重。近年来，能耗问题已成为服务质量研究中的热点问题，是服务设计和运行中需要关注的重要因素。

服务质量研究的一个重要理论基础是服务质量的评价和优化。它可以定量描述服务质量多个维度的属性指标，评价各种服务设计方案或运行策略在不同维度指标上的优劣程度，并从理论上指导服务系统的构建和服务运营方案的选择。目前，大部分的服务质量优化工作都是以数据作为驱动，根据对已有服务或服务质量的测量结果，预测未知方案的服务质量表现，或者直接对新方案实现后进行测量，进而选取最优的方案。在这种方法中，原始数据的获取需要对服务系统进行实现和部署，开销巨大；基于数据的预测受限于数据规模和质量，可能存在偏差，不能反映服务系统的本质行为。因此，需要从服务内部的特性出发，描述服务的动态行为，通过数学建模和分析方法，建立服务质量多维度属性的量化评价体系，进一步分析讨论模型参数和评价结果之间的量化关系，以便得到最优的系统参数或配置方案。随机建模分析方法在服务质量评价和优化方面提供了可行的新思路 and 新技术，为服务质

量的保障和优化提供了理论依据和支撑技术，这将是一个重要的充满前景的研究方向。

本书作者在基于随机模型的服务质量评价和优化领域进行了一系列深入而系统的研究工作。本书主要以马尔可夫模型、排队模型等为基础，以最优化方法、马尔可夫决策等为技术支撑，探讨服务质量的建模和优化方法，并结合服务计算中的具体问题进行探讨和验证。本书绝大部分内容取材于作者近期在国际一流学术期刊和会议上发表的论文，全面而系统地展示了很多新的研究成果和进展。

作者的研究工作得到了国家自然科学基金项目（No. 61502043）、北京市自然科学基金项目（No. 4162042）和北京市优秀人才培养资助项目（No. 2015000020124G082）的资助。

由于作者水平有限，加之基于随机模型的服务质量评价和优化的研究仍处于不断发展之中，书中存在疏漏和不足之处在所难免，恳请专家、读者予以批评指正。

黄霁崑

2017 年 4 月于北京

目 录

前言

第 1 章 绪论	1
1.1 研究背景	1
1.1.1 服务计算	1
1.1.2 服务质量的评价与优化	3
1.2 研究内容	3
1.3 研究难点与创新点	4
1.4 章节组织结构	5
第 2 章 相关研究综述	7
2.1 服务质量的评价指标	7
2.1.1 性能指标	7
2.1.2 可信赖性指标	7
2.1.3 能耗指标	8
2.2 服务计算的评价方法	8
2.2.1 模型方法	8
2.2.2 测量方法	11
2.2.3 预测方法	11
2.3 服务计算中的多属性问题	12
2.3.1 多属性之间的相互关系	12
2.3.2 多属性优化	12
2.4 讨论与总结	13
第 3 章 基于排队模型的性能评价	15
3.1 服务和服务系统的性能模型	15
3.1.1 原子服务的性能模型	15
3.1.2 服务系统的性能模型	16
3.2 模型求解与量化分析	17
3.2.1 马尔可夫排队模型的求解	18
3.2.2 半马尔可夫排队模型的求解	19
3.3 性能优化举例	20
3.3.1 服务节点的资源管理	20
3.3.2 分层系统的任务调度	22

3.4	案例分析	23
3.4.1	数据集介绍	23
3.4.2	原子服务的模拟实验	24
3.4.3	服务系统的模拟实验	26
3.5	本章小结	27
第4章	基于半马尔可夫模型的可信赖性评价	29
4.1	服务计算系统的可信赖性建模	29
4.1.1	基于半马尔可夫过程的系统状态建模	29
4.1.2	可信赖性形式化定义	31
4.2	模型分析与可信赖性评价	32
4.2.1	稳态概率分析	32
4.2.2	平均失效时间分析	33
4.3	可信赖性属性之间的相互关系	37
4.4	服务选择和服务组合的可信赖性建模与分析	41
4.4.1	基本框图模式的可信赖性分析	41
4.4.2	基本服务模式的可信赖性分析	42
4.5	案例分析	44
4.5.1	系统简介	44
4.5.2	可信赖性分析	44
4.5.3	敏感度分析	46
4.6	本章小结	48
第5章	基于随机游走模型的服务质量排序评价	49
5.1	协同分布式服务评价框架	50
5.2	服务评价排序模型	50
5.2.1	基于比较的概率模型	51
5.2.2	全局排序的随机游走模型	51
5.2.3	随机游走的马尔可夫模型	52
5.3	协同服务评价算法	52
5.3.1	算法概述	52
5.3.2	比较聚合方法	53
5.3.3	求解马尔可夫链算法	53
5.4	算法分析	55
5.4.1	误差分析	55
5.4.2	时间复杂度分析	55
5.5	案例分析	56
5.5.1	数据集简介	56
5.5.2	评价指标	57
5.5.3	实验结果	57
5.6	本章小结	59

第 6 章 基于折扣马尔可夫决策的性能—能耗综合优化	61
6.1 问题描述	61
6.2 服务系统的性能—能耗评价模型	62
6.2.1 原子服务的性能模型	62
6.2.2 嵌入离散时间马尔可夫回报模型	64
6.2.3 模型分析的更新技术	65
6.3 基于马尔可夫决策的动态功耗控制	66
6.3.1 问题描述和求解方法	66
6.3.2 马尔可夫决策对马尔可夫回报优化的有效性证明	68
6.4 分布式的动态服务选择与功耗控制综合优化	68
6.4.1 优化框架	68
6.4.2 优化算法	69
6.5 案例分析	71
6.6 本章小结	76
第 7 章 基于平均时间马尔可夫决策的可信赖性—能耗综合优化	77
7.1 服务系统的可依赖性—能耗评价模型	78
7.1.1 可信赖性模型	78
7.1.2 能耗模型	79
7.1.3 嵌入离散时间马尔可夫模型	79
7.2 基于回报值等价的状态聚合方法	80
7.2.1 可信赖性模型中的状态聚合	80
7.2.2 能耗模型中的状态聚合	82
7.3 基于马尔可夫决策的可信赖功耗控制算法	84
7.3.1 优化建模与算法	84
7.3.2 最优性证明	85
7.4 案例分析	86
7.5 本章小结	87
第 8 章 总结与展望	89
8.1 总结	89
8.2 研究展望	90
参考文献	93

绪 论

1.1 研究背景

1.1.1 服务计算

近年来，互联网的发展与普及为计算机软件的设计模式带来了新的思路与挑战。随着软件工程理念和技术的不断进步，传统的基于硬件和系统的架构模式逐渐转变为面向服务的设计模式。服务业已成为 IT 系统的基础组成部分^[1]。一种新兴的计算模式——服务计算（Services Computing）应运而生，并被越来越广泛地应用于各个领域。这里，服务（Service）是指服务提供商与用户之间为实现特定的商业目标或解决方案而形成的一种契约关系；服务计算则是指服务提供商将特定的功能封装成遵循统一标准的、交互式的计算机程序，并将其以服务的形式提供给不同商业领域中的多个用户使用的一种计算模式^[2]，它构成了连接商业服务与信息服务的桥梁。服务计算带来了商业模式和计算模式的转型。对服务计算的研究已成为目前学术界和工业界的研究热点，形成了涉及服务科学与信息技术的交叉学科。

服务计算是在预先定义好的标准框架下对服务过程进行设计、操作、管理和优化，以满足日益增长的动态的业务需求^[3]。它涉及面向服务的体系结构（Service Oriented Architecture, SOA）、Web 服务（Web Service, WS）、网格计算（Grid Computing）、云计算（Cloud Computing）等多项技术，涵盖了服务的整个生命周期的所有阶段，包括服务咨询与企划（Consulting and Strategic Planning）、服务契约（Services Engagement）、服务发布（Services Delivery）、服务运营（Services Operation）、服务计费（Services Billing）、服务管理（Services Management）等，由此也产生了服务标准制定、架构设计、服务发现、服务推荐、服务选择、服务组合、监控与管理、服务质量评价和保障等一系列科研问题^[4]。

服务计算系统是基于 IT 技术，为实现特定的业务服务需求而构建的计算机系统，为服务的持续性工作提供硬件保障和技术支撑^[4]。一个服务计算系统包括 3 层结构，底层由基础设施构成，基于服务计算技术构建的大规模数据中心是提供服务的基础设施，它可以看做是一个通过网络互联的大型服务器集群，承载着各种应用和服务^[5]；第 2 层是管理模块和中间件，包括虚拟机管理、服务质量管理、服务组合、服务监控、计费系统等模块，它将服务层的应用和底层的硬件基础设施联系起来，并具有松耦合的特点，使整个系统具有更灵活的特

性；第3层是服务层，面向用户，通过统一的标准化接口，为用户提供满足需求的各种类型的服务。借鉴典型的服务计算实例——云计算系统作为参考^[6]，结合服务计算普遍的特性，可以得到典型服务计算系统架构示意图，如图 1.1 所示。

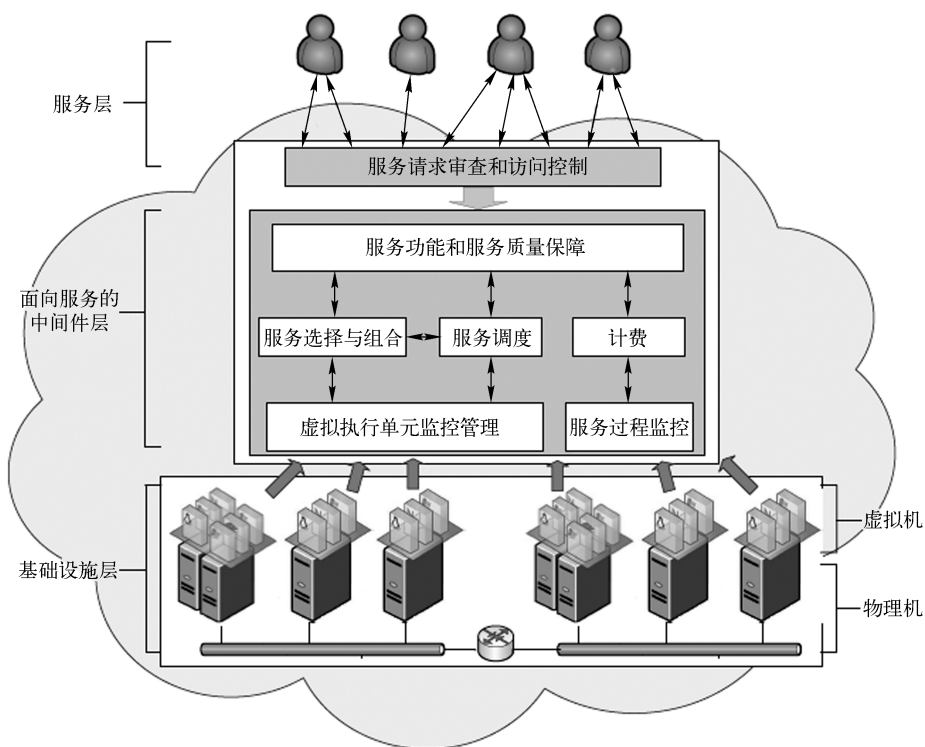


图 1.1 典型服务计算系统架构示意图

随着服务计算的逐渐普及，用户需求的不断提高，服务计算及服务计算系统呈现如下 3 个显著特点。

1) 动态性 为了满足用户日益增长的多样化需求，提高服务的利用率，服务之间可以通过服务组合^[7]的形式调用已有的服务，并且相互协同^[8]，从具有简单功能的服务构建复杂的服务工作流。服务组合和服务协同使得服务的过程具有高度动态性，一个服务过程可能会涉及一系列的服务和系统组件。

2) 松耦合 服务计算中，通过服务描述语言（Web Services Description Language, WSDL）^[9]定义统一的服务接口，降低了上层服务与下层实现机制的依赖程度（即耦合性）。另一方面，虚拟化是服务计算系统构建中的常用技术^[10]，硬件资源的虚拟化使得软件级的服务实现与系统底层硬件实现了松耦合，服务计算中的上层服务与底层硬件之间不再具有固定的对应关系。虚拟机迁移^[11]和虚拟资源共享^[12]机制则进一步增加了耦合的灵活性，并提高了资源利用的效率。

3) 大规模 在服务计算中，服务被部署在分布式的硬件基础设施上。随着服务种类和数目的不断增多，基础设施的规模也在不断增大^[2]。服务计算中的底层数据中心往往包括成千上万的服务器和大量的网络设备^[5]。服务和硬件设备的大规模特性，为系统的设计、服务的部署、资源的管理带来了严峻的要求和巨大的挑战。

1.1.2 服务质量的评价与优化

服务计算兴起之初,绝大多数的研究主要关注于服务计算功能性(Functional)需求的满足,研究内容包括服务标准的制定、服务流程的设计和服务系统的实现等。随着服务计算被越来越广泛地应用,各种类型的应用对系统和服务提出了越来越高的需求,其中非功能性(Nonfunctional)的需求,即服务质量(Quality of Service, QoS),逐渐被关注和研究^[13]。这里,服务质量可以包括多种指标,每类指标又包含多个属性。其中,第一类是传统的性能指标,它反映了系统处理能力或服务效率,包括响应时间、吞吐率、利用率等属性;第二类是服务的可信性^[14],它描述了服务计算系统避免或容忍组件或服务失效,以持续提供可用服务的能力,包括可靠性、可用性、可维护性、完整性、保险性等;第三类是服务过程中所消耗的能源或系统运行的能耗效率,即能耗与功率^[15]。

目前,在很多服务过程中,用户均对服务提供商提出了QoS的需求,这些需求往往通过规范化的服务等级协议(Service - Level Agreement, SLA)得以体现^[16]。它是在一定开销下为保障服务的性能和可靠性等QoS指标,服务提供商与用户之间定义的一种双方认可的协定。SLA协议已成为服务计算QoS保障中的一个重要方面。

在服务设计和系统实现过程中,为了保障服务的QoS,首先需要研究系统配置、负载对QoS指标的影响,从而给出QoS的准确评价;其次,需要对服务计算系统进行优化,选择最优的系统配置和服务解决方案,以适应不同的环境与需求。优化过程覆盖服务系统的整个生命周期,涉及多个学科,如运筹学、复杂系统建模、系统工程等^[2]。服务质量的评价与优化是服务计算领域中的一个热点研究问题。

1.2 研究内容

性能评价(Performance Evaluation)^[17]是指对系统的动态行为进行分析和优化,包括对实际系统的行为进行建模,对系统性能指标进行测量,对已有系统的性能缺陷或瓶颈进行改进,根据性能需求对系统实施方案进行设计或选择。性能评价是计算机系统和计算机网络研究与应用的重要理论基础和支撑技术,也是当今通信和计算机科学领域的重要研究方向^[18]。

基于随机模型的方法是性能评价的重要手段,其核心思路是对要评价的服务或系统建立一个适当的数学模型,然后根据模型参数求解评价指标。模型参数的确定往往依赖于对实际评价对象的测量结果或其参数的估计^[19]。基于随机模型的评价方法可以清晰地描述各种因素之间的关系,并且可以应用于尚未存在的系统的性能预测,普适性好,评价成本低,因而在服务计算的QoS评价中被广泛使用。根据模型分析结果,可以利用数学方法或计算机方法调整系统参数或配置策略,在保证用户得到约定或希望的服务质量的前提下,使资源利用率最大化,提供服务质量最优的服务。

随着服务计算中QoS受到越来越广泛的关注,如何对服务计算系统进行评价、比较和改进,是服务计算中一个重要的问题。性能评价理论与技术逐渐发挥出其巨大的价值,它包括如下5个方面。

1) 多维度的评价指标 考虑服务计算的特点和需求,建立完整统一的多维度、多属性评价指标体系,可以全面、客观地对系统进行评价。

2) 随机评价模型 根据评价指标,建立系统的随机理论模型,可以对系统的动态行为、资源管理和任务调度策略进行形式化的描述。

3) 量化分析 在模型基础上,对系统进行量化分析,给出指标的形式化解析解,以便对不同的系统进行分析和比较。

4) 属性之间相互关系的研究 各属性之间存在相互影响的关系,对其进行深入研究可以完善整个评价和优化的理论体系,并为多指标优化提供理论上的依据。

5) 多属性综合优化 研究服务计算系统的多属性综合优化方法,对系统设计改进和最优控制具有重要的指导意义。

本书针对服务计算背景下的性能评价展开研究,从性能、可信赖性和能耗3个维度,考虑每个维度下的多个属性,建立服务计算的随机评价模型;研究模型的分析求解方法,给出各属性的形式化的量化表达;研究属性之间的相互关系,结合随机评价模型给出其数学表达,用以指导系统优化设计;在定量分析的基础上,研究优化方法,结合随机模型理论,给出服务质量动态保障机制和优化策略。

1.3 研究难点与创新点

服务计算环境的动态性、复杂性、大规模、资源的异构性等,给建模、评价和优化工作带来了一系列困难和挑战。本书的研究工作分别针对以下难点,进行了创新性的研究。

1) 服务质量的多维度、多属性的评价指标体系 服务计算中,对性能、可信赖性和能耗3个维度均提出了严格的要求,每个维度又都包含多个属性,从而构成了一个多维度、多属性的评价指标体系。如何全面、客观、普适地建立评价指标体系,用以对服务计算中的计算流程、系统实现进行分析和比较,是一个研究难点和热点。

【创新点1】 本研究从不同层面研究系统建模的理论框架和评价体系,分析和解决相关科学问题,建立服务计算系统中性能、可信赖性和能耗的多维度、多属性评价指标体系。在此基础上,进一步研究多指标之间的相互关系,完善多属性评价的理论框架,为改善服务系统提供科学支撑。

2) 服务计算的建模方法 服务计算中的动态性、松耦合和大规模的特点,使得传统的基于组件的建模方法面临了巨大的困难。用户得到的服务和底层硬件不再具有固定的对应关系,单纯地进行系统组件的建模和分析难以反映服务的特性。服务计算中亟需新的建模思路和方法,以有效地对系统行为和特性进行描述。

【创新点2】 本研究从面向服务的角度出发,研究服务计算的特性,建立其评价模型。该面向服务的建模方法从服务层出发,自上而下对服务计算系统进行描述和分析,对其进行综合度量,有利于实时监控系统状态,深入理解系统特性,指导建立相应的保障机制,并丰富建模理论。

3) 服务计算评价模型的分析方法 服务计算的大规模特性,使得分析评价的过程中可能会遇到状态空间过大,模型求解过程过于复杂,服务评价成本过大等问题。朴素的模型求解和评价方法在大规模服务的分析过程中可能不再有效。高效的服务分析方法和评价技术,是服务计算中亟待解决的问题。

【创新点3】 本研究从服务计算的模型出发,针对其马尔可夫模型和半马尔可夫评价模

型,分别给出了状态聚合方法。从理论上可以严格证明,聚合后的模型和原模型在评价指标上等价。这些方法的提出,可以有效降低模型状态的规模,并能更清晰地反映评价对象的特性,直观描述系统的动态行为。

【创新点4】本研究针对大规模服务评价中的服务排序问题,提出了高效的分布式服务评价技术,及其相关的排序聚合方法。该方法可以显著地降低服务评价的成本,并且将误差控制在可接受的范围内。分布式的服务评价技术可以为大规模的服务选择、服务组合、服务推荐等提供理论支持和技术保障。

4) 服务计算多属性动态优化方法 服务计算的动态性对其参数配置和流程优化带来了挑战。考虑服务计算中对多维度多属性的需求,如何在理论上保证优化方法的最优性,同时在技术上保证算法的效率,是本研究中的核心问题。

【创新点5】本研究采用马尔可夫决策过程对服务计算中的动态优化进行形式化建模。在理论上证明了马尔可夫决策过程与服务计算的评价模型(马尔可夫回报模型)优化的一致性,并量化地给出了二者在优化目标上的对应关系。在技术上,借鉴马尔可夫决策中的前沿技术和相关算法,给出了高效可行的优化方法。

【创新点6】为了在大规模待选服务中选取最优解,本研究提出了基于多 Agent 技术的分布式协同优化方法,给出了优化框架及其相关的优化决策算法。利用 Agent 间相互通信和协作的特性,将问题分而治之,并行地寻找局部最优解,进而从局部最优解中找到可行的全局最优解。

1.4 章节组织结构

本书围绕性能、可信赖性和能耗3个维度开展了随机建模、量化分析和优化方法研究,并针对服务排序这一类特殊的评价问题,进行了普适性的建模理论和评价方法的探讨。

全书内容共分为8章。其中,第1章为绪论;第2章介绍相关研究工作,对服务质量的评价指标、评价方法和多属性优化等方面的已有研究进行综述;第3章研究服务计算系统的性能评价方法,基于排队论建模理论,提出了原子服务和性能模型,给出了模型的量化分析方法,并介绍了资源管理和任务调度等典型优化问题的解决方法;第4章研究服务计算系统的可信赖性评价,提出了面向服务的可信赖性评价模型,给出了可信赖性各属性的分析方法,讨论了可信赖性属性间的相互关系;第5章研究服务排序评价的整合测量技术,旨在大规模服务环境下,针对排序评价结果,有效地对评价结果进行聚合,从而减小评价的开销,并介绍了基于比较的排序模型,给出了模型分析算法,通过理论分析和实验验证证明了模型和算法的有效性;第6章研究服务计算中性能和能耗的综合评价和优化,提出了基于马尔可夫回报模型的系统建模方法,给出了性能和能耗的量化分析,以及基于回报值的综合优化框架和算法;第7章综合考虑性能收益、可信赖性和能耗,给出了普适的评价模型,介绍了基于平均时间马尔可夫决策理论的综合优化方法,并在理论上证明了优化方法的有效性;第8章总结全文,并对未来的研究方向进行展望。

相关研究综述

随着服务概念的推广，服务计算逐渐成为一个活跃的研究领域。服务质量的评价和优化，是服务计算中的一个核心问题，关系到用户需求的满足、服务计算系统的优化设计和有效管理，因此引起了相关领域专家和学者们极大的研究兴趣。

2.1 服务质量的评价指标

服务质量的评价是对服务和系统优劣的客观量化描述，是对系统设计、服务管理方案进行比较和选择的基准。随着服务计算的兴起和广泛应用，用户多样化的需求对服务质量的多个维度、多个指标提出了严格的要求。目前，服务质量的评价可以从 3 个维度展开，即性能、可信赖性和能耗。

2.1.1 性能指标

性能反映了服务的能力或效率，是服务质量中的一个重要方面。性能指标是一个综合性的指标，包括响应时间（Response Time）、吞吐率（Throughput）、利用率（Utilization）等属性。

响应时间是从服务请求发出到执行完成的时间，包括请求发送延迟、等待时间、服务执行时间等。在服务计算中，响应时间是性能的代表性指标，与服务器负载、服务调度策略、数据带宽等系统参数有密切的联系^[20]。

吞吐率是指系统单位时间内可以完成的服务的数目，是评价服务计算系统最大服务能力的参考指标^[21]。

利用率反映了服务或系统资源被使用的情况，是整个服务过程中该服务或承载服务的软/硬件资源被使用的比率。服务和系统资源的利用率是系统和服务管理、资源分配、任务调度等过程的一个关键因素，并直接关系到服务提供商的利益，在服务计算的评价和优化过程中需要加以关注^[22]。

2.1.2 可信赖性指标

随着服务计算的广泛应用，服务种类的增多对服务质量提出了更高的需求。除传统的性能指标外，用户对持续服务、可靠运行的需求成为了关键需求，尤其是在一些特定的服务应

用中,如交通管制、医疗卫生监护、军事应用等。可信赖性 (Dependability) 是指一个系统避免发生严重的服务失效或过于频繁超出可接受范围的故障的能力^[14,23],它已逐渐成为了学术界和工业界的研究热点。

可信赖性是一个综合性的概念,它包含如下 5 个不同的属性^[14,24]。

- 1) **可用性** 是指当用户需要时,系统可以向用户提供正常服务的能力。
- 2) **完整性** 是指抵御不适当的系统或数据修改的能力。
- 3) **保险性** 是指服务在生命周期内,不会对用户和环境造成灾难性后果的能力。
- 4) **可维护性** 是指系统进行修复和容错的能力。
- 5) **可靠性** 是指服务能够持续正常工作的能力。

在服务计算系统的生命周期中,其可信赖性面临的威胁包括 3 类,即缺陷 (Fault)、错误 (Error) 和故障 (Failure);而增强可信赖性的手段包括缺陷的预防 (Fault Prevention)、容忍 (Fault Tolerance)、修复 (Fault Removal) 和预测 (Fault Forecasting)^[14]。

2.1.3 能耗指标

近年来,计算系统中的能量消耗受到了越来越多的关注。有报告指出,计算机服务器的能耗开销呈逐年增长的趋势^[25]。在 Web 服务、云计算、数据中心等大规模服务计算系统中,能源消耗的问题更加严峻^[26]。2006 年,仅在美国境内,IT 基础设施的运营和维护就消耗了 $610 \times 10^8 \text{ kW} \cdot \text{h}$ 的电量,造成了超过 3200 万美元的开销^[27]。2010 年,仅 Google 一家公司就消耗了 $22.6 \times 10^8 \text{ kW} \cdot \text{h}$ 的电量^[28]。截至 2011 年,为了承载互联网上广泛发布的各类服务,底层数据中心消耗的电量占据了全球耗电总量的 1.3%;据推测,该比例将在 2020 年增长至 8%^[29]。能耗问题已经成为服务计算设计和维护中不可或缺的关键因素^[15,30]。

能耗指标主要包括两个属性,即功率 (或功耗, Power) 与能耗 (Energy)^[26]。系统的功率 $P(\tau)$ 是指系统在运行过程中的某个时间 τ 消耗能量的速率,而能耗是指一段时间 t 内系统消耗的总能量。功率与能耗之间的对应关系为

$$E(t) = \int_0^t P(\tau) d\tau \quad (2.1)$$

2.2 服务计算的评价方法

针对多维度的评价指标,如何得到指标的数值,是服务计算评价中需要解决的核心问题之一。服务计算的评价方法可以分为 3 类,即模型方法、测量方法和预测方法。

2.2.1 模型方法

基于模型的方法是性能评价的重要手段,其核心思路是对要评价的服务或系统建立一个适当的模型,然后根据模型参数求解出评价指标。这里,模型参数的确定往往依赖于对实际评价对象的测量结果或对其参数的估计^[19]。基于模型的评价方法可以清晰地描述各种因素之间的关系,并且可以应用于尚未存在的系统的性能预测,普适性好,评价成本低,因而在服务计算的绩效评价中被广泛使用。

1. 马尔可夫模型

马尔可夫模型 (Markov Model) 是性能评价常用的模型工具。它利用马尔可夫过程

(Markov Process) 对系统的动态行为进行描述。马尔可夫建模方法要求所描述的系统或服务的状态变化具有无后效性 (或称为马尔可夫特性), 即对于系统状态变化的随机过程 $\{X(t), t \in T\}$, 如果对于任意的时间参数 $t_0 < t_1 < \dots < t_n < t$, 在 $X(t_0), X(t_1), \dots, X(t_n)$ 已知的情况下, $X(t)$ 的条件分布只与 $X(t_n)$ 状态相关, 即

$$\Pr\{X(t) \leq x \mid X(t_0) = x_0, X(t_1) = x_1, \dots, X(t_n) = x_n\} = \Pr\{X(t) \leq x \mid X(t_n) = x_n\} \quad (2.2)$$

一般地, 在性能评价研究中可以假定系统的行为不依赖于观测时间, 符合时齐性 (Time Homogeneous), 即认为马尔可夫过程的条件概率分布函数与观察起始时刻无关^[19]。在评价中, 可以任意选取时间轴起点 $t_n < t$, 时齐性保证下式始终成立:

$$\Pr\{X(t) \leq x \mid X(t_n) = x_n\} = \Pr\{X(t - t_n) \leq x \mid X(0) = x_n\} \quad (2.3)$$

具体地, 在性能评价中使用的马尔可夫模型也可进一步细分为 3 类, 即马尔可夫链 (Markov Chain)、马尔可夫回报模型 (Markov Reward Model) 和半马尔可夫模型 (Semi-Markov Model)。

1) 马尔可夫链 考虑到绝大多数计算机系统的状态空间均为离散的, 这样的马尔可夫过程称为马尔可夫链 (Markov Chain, MC)。根据时间参数 t 的取值范围 T , MC 又可以进一步分为离散时间马尔可夫链 (Discrete-Time Markov Chain, DTMC) 和连续时间马尔可夫链 (Continuous-Time Markov Chain, CTMC) 两种。

马尔可夫链被广泛用于计算机系统的性能建模。文献 [31] 采用 CTMC 对云计算中的服务器进行了建模, 将服务器缓存中的虚拟机迁移请求数目定义为 CTMC 的状态, 综合考虑虚拟机迁移请求到达和服务过程, 应用马尔可夫模型理论和验证技术对系统性能进行评价。文献 [32] 针对大规模网格服务系统建立了系统状态转移的 DTMC 模型, 描述了系统在大规模请求到达或出现故障情况下的动态响应和状态变化, 用以评价系统的可用性和可靠性属性。文献 [33] 对服务计算中的虚拟机集群进行了分析, 根据虚拟机状态建立了 CTMC 模型, 通过 CTMC 稳态概率的求解, 给出了系统可用性的量化分析。文献 [34] 建立了服务器状态的 DTMC 模型, 研究系统的能耗状态, 用以实现性能、可靠性和能耗平衡。

2) 马尔可夫回报模型 在马尔可夫链中, 若对每个状态定义一个回报值, 代表系统在利润、性能、可靠性等方面的收益, 则该 MC 称为马尔可夫回报模型 (Markov Reward Model, MRM)。与 MC 类似, MRM 同样可以分为离散时间和连续时间两种时间维度。

回报值的引入为性能评价提供了巨大的便利。MRM 可以将模型结构和系统需求紧密联系起来, 并且可以和马尔可夫决策理论中的优化过程相结合, 是性能、可靠性等分析评价的有力工具^[35]。文献 [36] 提出了生产服务系统的离散时间 MRM 建模方法, 用以评价系统的性能, 在该模型中, 状态回报值可以具有随机性, 以适应实际系统中更为复杂的情形。文献 [37] 采用连续时间的 MRM 对服务共享系统的状态转移进行建模, 通过模型分析给出了系统可用性的评价方法。文献 [38] 从离散时间和连续时间两个维度建立了服务组合和系统资源的 MRM, 对服务的可靠性进行了高效而准确的分析。

3) 半马尔可夫模型 半马尔可夫模型是对经典马尔可夫模型的推广。考虑一个有限状态、连续时间、齐次的随机过程 $\{X(t), t \geq 0\}$, 在时刻 $t_0 < t_1 < \dots < t_n$ 状态发生变化, 并且在时间间隔 $[t_n, t_{n+1})$ 有值 Y_n , 满足:

$$\begin{aligned} \Pr\{Y_{n+1} = j, t_{n+1} - t_n \leq \tau \mid Y_0 = y_0, Y_1 = y_1, \dots, Y_n = i; t_0 = \tau_0, t_1 = \tau_1, \dots, t_n = \tau_n\} \\ = \Pr\{Y_{n+1} = j, t_{n+1} - t_n \leq \tau \mid Y_n = i\} = H_{ij}(\tau) \end{aligned} \quad (2.4)$$

在该过程中, 状态逗留时间的分布函数 $H_{ij}(\tau)$ 可以是任意分布的, 并且可以依赖于当前状态和下一个状态。这样的随机过程 $\{X(t), t \geq 0\}$ 称为半马尔可夫过程 (Semi - Markov Process, SMP)。可以证明, 随机序列 $\{Y_n, n \geq 0\}$ 是一个 DTMC, 该过程被称为过程 $\{X(t), t \geq 0\}$ 的嵌入马尔可夫链 (Embedded Markov Chain, EMC)。

半马尔可夫模型一般被应用于特殊系统的状态转移建模。当系统状态转移过程中各状态的逗留时间不再服从指数分布时, 可以采用 SMP 进行系统动态行为的描述。文献 [39] 中针对软件系统中安全事件和修复过程的时间分布非指数分布的情况, 建立了系统状态转移的 SMP 模型, 给出了模型的稳态概率分析, 从而得到系统安全性属性的量化表达。文献 [40] 中应用半马尔可夫模型分析了 SOA 系统的安全容侵特性, 给出了系统安全性的平均失效时间 (Mean Time to Security Failure, MTTSF) 和平均修复时间 (Mean Time to Repair, MTTR)。文献 [41] 中利用半马尔可夫模型对 IP 网络流量进行了描述, 建立了网络流量的特性和状态间的转换关系, 并通过实际数据分析验证了模型的正确性。

2. 排队模型

排队论 (Queueing Theory), 又称为随机服务系统理论, 是性能预测、性能分析和性能评价中的一个有力手段^[19]。它可以描绘一个系统对群体提供服务时, 系统所呈现的各种状态, 包括用户请求的到达、服务、排队、拥塞等过程, 进而可以对系统的性能进行评价。

在排队模型中, 需要对到达规律、服务时间分布、服务器数量等进行预先规定。经典排队模型可以用 A/B/n/S/Z 格式表示。其中, A 代表请求到达的规律; B 代表服务时间的分布; n 代表服务器的数目; S 代表队列容量的大小; Z 代表服务规程。当队列容量为无穷大, 服务符合先到先服务 (First Come First Serve, FCFS) 的服务规程时, 该排队模型可简化表示为 A/B/n 的形式。最典型的排队模型是 M/M/1 模型, 代表请求到达为泊松过程 (Poisson Process), 服务时间服从指数分布 (Exponential Distribution), 系统中只有一个服务器。在经典排队模型基础上, 还发展出了排队网络、随机网络演算等性能分析工具。

对于某些常见的排队模型, 可以建立其对应的马尔可夫链, 从而进行进一步的求解和分析。例如, M/M/1 排队模型对应的马尔可夫链是一个生灭过程 (Birth - Death Process), 即所有的一步转移只发生在相邻的状态之间, 分别对应请求到达和服务过程。多个 M/M/1 队列组成的排队网络可以对应为一个多维马尔可夫链, 该马尔可夫链状态间的转移速率由排队网络中的到达速率、服务速率和队列的组成关系共同决定。M/G/1 模型可以对应到一个半马尔可夫过程, 因而结合其服务时间分布对 SMP 的嵌入马尔可夫链进行分析, 即可得到队列的性能指标。

排队模型被广泛用于服务计算系统的性能分析, 以及服务生命周期内端到端服务质量的评价和保障。文献 [31] 建立了云计算系统中单服务器的 M/M/1 排队模型和多服务器的排队网络模型, 针对虚拟机迁移过程建立了系统的性能模型。文献 [42] 采用 M/M/1/k 排队模型对云计算中的虚拟机进行建模, 将云计算系统中的多个虚拟机组成排队网络, 用以分析端到端的服务质量。文献 [43] 建立了计算系统的 MMPP/M/1/k 和 M/M/1/k 模型, 给出了响应时间和利用率的评价方法。文献 [44] 利用 G/G/1/FCFS、M/G/1/PS 和 M/G/ ∞ 排队网络模型对 Web 服务组合中的服务时间等性能指标进行分析, 提出了互联网环境下 Web 服务组合的性能预测方法。

3. 组合模型

组合模型主要用于系统可信赖性分析, 是除马尔可夫模型外的另一个重要建模工具^[45]。

组合模型的建模思路是,首先分析系统的组成结构及其相互关系,然后以系统组件为单元,将系统组成抽象为并行、串行等连接关系,进而分析系统整体的可信赖性属性。与马尔可夫模型相比,组合模型较为直观和简洁,分析方法简便、易行;但组合模型描述能力较弱,难以完成复杂时间维度的分析。

在可信赖性分析中,最常见的组合模型为可靠性框图 (Reliability Block Diagram, RBD)^[46],在系统中存在复杂的层次式组件构成或服务组合时,可以根据组件之间的相互关系由子部件的可靠性得到整体的可靠性表达^[47]。失效树 (Fault Tree, FT) 是另一种常见的组合模型,它从系统中可能遇到的故障出发,研究故障发生的原因,进而评价系统的可信赖性属性^[48],该模型思路也可用于安全评价方面,如攻击树 (Attack Tree, AT) 等组合模型^[49]。

2.2.2 测量方法

测量方法是指通过一定的测量设备或程序,直接从计算机系统测得各项性能指标或与之密切相关的度量,然后经过一些简单的计算求出需要的性能指标^[19]。测量方法是最为直接的性能评价方法,也是最基本的方法,其他方法在一定程度上需要依赖于它。例如,模型方法中的系统参数往往需要统计测量才能得到。测量过程中得到的结果,如系统到达服从的分布特性、状态转移之间的时间分布等,对模型的建立具有重要的指导作用。

性能测量在服务计算中广泛应用,这为理解系统特性,了解其内在规律提供了参考。文献 [50] 针对现有的数个典型云计算系统,在不同的负载下测量了云服务的响应时间、系统利用率等性能指标,并根据云计算中的计算需求提出了系统改进方案和建议。文献 [51] 提出了一种服务器虚拟化的测试基准,通过研究虚拟化数据中心中的测量数据,建立了性能评估模型。文献 [52] 提出了一种分布式 Web 服务质量评价方法,使用该方法对互联网上已有的 5825 个真实的 Web 服务进行了测量,对服务的响应时间、吞吐率、可靠性等进行了评价。

2.2.3 预测方法

目前,互联网上发布、运行着成千上万的服务,随着服务数目的增多,相似功能的服务数量逐渐增大,大规模服务集合中的服务比较和选择成为一个亟待解决的问题。无论采用何种方法对服务质量进行评价,即使单个服务的评价效率很好,但当服务数量增长到一定规模时,对如此大量的服务的服务质量评价和整合将面临巨大的挑战。如何高效地完成服务质量评价,降低评价的开销,已被广泛关注和研究。

针对这一挑战,除模型方法和测量方法外,基于预测的评价方法应运而生并被广泛使用。其主要思路是,根据已有的一些测得或计算得到的服务评价数据,以及未知服务的特性和周围环境,对其服务质量进行预测,从而避免对所有的服务进行评价。文献 [53] 利用已知服务的历史故障数据,根据用户间的相似度,对未知用户环境下的未知服务进行可靠性预测;该文献还提出了用户协同的故障数据共享机制和可靠性评价聚合方法,以进一步提高预测的准确度和效率。文献 [54] 利用用户间的邻接关系,提出了服务质量的协同过滤预测方法,对服务的响应时间和吞吐率两个性能指标进行了讨论。文献 [55] 进一步考虑了协同过滤预测方法中的数据可信度的问题,给出了数据集的分析评价方法和数据筛选方法,以提高预测准确度。

2.3 服务计算中的多属性问题

针对服务的多方面需求,考虑到不同属性之间存在相互关联和影响,相关研究已注意到服务计算中的多属性问题。这些研究主要包括两个方面:一是研究多属性的综合评价模型与属性之间的相互关系,二是研究服务计算中多属性的综合优化问题。

2.3.1 多属性之间的相互关系

服务计算的评价中涉及多个维度的评价指标。这些指标属性在系统中并非独立存在,它们之间存在一定的相互关系,指标之间可能相互影响。对属性相互关系的研究,有助于深入了解服务计算的指标特性,为多目标优化提供一定的理论指导。

首先,同一维度内的指标存在一定的相互关系。在性能维度内,响应时间和吞吐率存在一定的相互关联(大致呈反比关系),因而在对这两个属性进行综合优化时,可以用 Power 公式作为二者的综合评价函数,如式(2.5)所示^[56]。在可信赖性维度内,可靠性和可用性之间也存在相互影响关系,文献[57]给出了服务可靠性和可用性相互关系的量化表达。

$$\text{Power} = \frac{\text{吞吐率}^\alpha}{\text{响应时间}} \quad 0 < \alpha \leq 1 \quad (2.5)$$

其次,不同维度的指标之间在一定情况下也会相互关联。例如,性能指标与可靠性之间存在一定的相互影响。系统组件的失效将影响该系统所承载服务的服务质量;同时,在系统故障恢复过程中,系统无法对服务进行响应和处理,使得服务请求处于等待状态,进而增加了响应时间^[58]。有研究表明,任务负载情况将同时影响系统的性能和可靠性,这也从侧面揭示了性能和可信赖性这两个维度指标之间的相互关系^[59]。另一方面,性能和能耗之间也存在紧密联系。系统不同的运行状态决定了系统的性能,也决定了系统的功率等级,性能与功率之间一般呈正相关的关系^[26,60]。此外,可靠性与能耗之间存在相互制约的关系。为了提高可靠性,常常需要增加系统部件备份,以防止部件故障影响服务能力,而增加备份必将带来额外的能源开销,二者之间的权衡是一个研究热点^[34]。

2.3.2 多属性优化

在众多评价指标存在的情况下,用户和服务提供商往往希望尽量多的指标属性能够更优,或者有些指标满足特定的需求^[61]。例如,用户可能希望服务的响应时间尽量小,可用性和可靠性尽量高;服务提供商可能以吞吐量大、可维护性高、能耗低为设计或运营目标。如何同时保证多个评价指标最优,即多属性优化(也称为多目标优化),是服务计算中资源管理、任务调度、服务组合、服务推荐等方面中的一个重要科学问题。

1. 评价函数方法

基于评价函数的优化方法是解决多属性优化问题的最常见的思路之一。该方法是利用一个评价函数,将多个优化目标映射成一个函数值,从而将多目标优化问题转化为单目标优化问题,再用经典或现代优化方法进行求解。

简单线性加权求和(Simple Additive Weighting, SAW)是广泛使用的一种指标聚合评价方法。它根据用户的需求,为不同的目标规定相应的权重,将所有的目标值进行线性加权,

用一个综合的效用函数来表达总体优化的目标。通常,需要将目标值按照其最大值/最小值进行归一化,将属性调整为0和1之间的值,去除目标属性的单位,进行规范化处理。基于SAW的优化方法目前被广泛应用于服务选择与组合^[62,63]、服务搜索和推荐^[64]等领域。

另一种评价函数为基于相乘或相除运算的方法。该方法主要应用于具有一定程度的正相关或负相关属性的评价。典型的例子为响应时间和吞吐率的综合优化问题,评价函数为Power公式;可以证明,在某个最优负载条件下,吞吐量和响应时间之比(即Power公式的值)可以达到最大值^[56]。

2. ε -约束方法

ε -约束的多属性优化模型是选取多个属性中的某一个属性作为新的优化目标,将其他属性通过增加界限值转化为约束条件,形成新的优化问题,从而将多目标优化转化为有多个约束条件的单目标优化问题^[65]。例如,在考虑服务计算系统中的资源管理时,可以将响应时间作为限制,将开销成本作为优化目标,实现资源的有效利用^[66]。在进行服务部署的优化设计时,可以将服务质量作为约束,在满足SLA的基础上,将最小化系统的运营开销作为优化目标,进而用线性规划等经典的单目标优化方法对研究问题进行求解^[67]。

3. 帕累托优化方法

帕累托(Pareto)是多属性优化中的常用模型^[68]。它改变了传统的选取一个最优解的思路,取而代之的是选取一组最优解(称为帕累托最优解)作为优化结果。选取的这组解具有帕累托最优的性质,即在帕累托解集外不存在某个解在每个优化目标上均比某个帕累托最优解更优。这样,得到的每个帕累托最优解均在一定程度上比其他解(包括其他帕累托最优解)更有优势。帕累托优化方法不用对属性值进行整合评价,可以适应各类指标,并且可以给用户更多的选择,适应用户的各种需求,因而比其他方法更具有普适性。

近年来,在服务计算中,帕累托优化方法被逐渐采用。文献[69]采用帕累托优化方法研究云计算中的资源管理和任务调度问题,提出了近似求解算法,以实现响应时间和开销之间的权衡和优化。文献[70]从帕累托最优的角度讨论了服务组合问题,定义了服务之间的比较关系,研究了服务组合过程中服务比较关系的变化情况,进而提出优化方法,得到帕累托最优的服务组合策略。

2.4 讨论与总结

服务计算的多属性评价和优化一直都是该领域内的一个热点核心问题。对评价指标、评价模型、评价方法、优化理论的研究,均对服务计算的设计与改进具有重要的意义。目前,该领域的专家、学者已对这些问题进行了初探,并取得了一定的成果。在服务基础设施、服务组合、服务推荐、服务管理等方面,评价和优化理论均显示出了令人兴奋的理论意义和应用价值。

但是,已有的研究还存在一定的空缺,需要进一步补充和完善。

(1) 在评价指标上,大部分研究只关注于部分属性,讨论不够全面。尤其是可信赖性的讨论,现有研究大都只关注于可靠性和可用性,而忽视其他属性。目前,在服务计算中尚未对可信赖性指标进行系统性的研究,缺乏全面的理论体系。另一方面,服务计算中对能耗的研究近年来才逐渐开展,对于功率和能耗的理论评价和优化技术还不够成熟,还有很大

的进一步研究的空间。此外，已有的模型方法往往只关注于某一个或两个属性的评价，缺乏能评价多个属性指标的普适的模型，具有局限性；服务计算中多属性的建模理论和方法尚待深入探讨。

(2) 考虑到属性之间存在复杂的相互关系，对其进行理论分析将为服务评价和优化提供理论支持。然而，现有的研究对这一方面的讨论还比较欠缺，指标体系的完整性和理论分析的深度均有待提高。另外，目前服务计算中尚无理论模型可以同时评价多个属性，从而无法在同一个层面对多个属性之间的相互联系进行分析，对属性之间相互关系的讨论的理论基础不强。

(3) 由于属性之间相互关联，在多属性优化的过程中不能简单地对其进行独立的讨论或简单的加权求和，而应结合多属性评价模型，考虑到属性之间的相互影响，给出优化方案。现有研究工作在优化过程中往往忽视了属性之间的相互影响，单独地对多个属性分别优化，缺乏理论依据。

因此，本书将从评价指标体系出发，建立具有普适性的综合评价模型；深入探讨属性之间的相互关系，包括同一维度内属性之间的相互关系，以及不同维度之间指标的相互关系。本书将给出可以评价多个属性的评价模型，从理论模型出发，研究属性之间的相互关系；基于模型给出优化路线，并在理论上证明优化方法的合理性和有效性。

基于排队模型的性能评价

性能是服务质量的最重要的指标维度之一。随着服务计算的应用和推广，尤其是对于军事、医疗、安防等应用场景，高性能或实时性服务的需求逐渐增多，响应时间、吞吐率等性能指标成为越来越重要的服务质量需求。如何对性能进行客观、准确的评价，使其作为衡量服务质量的重要标准，并且为系统优化和服务选择提供客观依据，是亟待解决的研究问题。

基于模型的方法是性能评价的一个有力的工具，它可以对服务或服务系统的动态行为进行数学描述，在系统设计阶段即可对服务质量进行预测性评价，从而避免实现和部署系统带来的巨大开销。其中，排队论（即随机服务系统理论）正是对服务系统进行数学描述的有效工具，在服务或服务系统的性能评价方面具有先天优势。因此，本章将利用排队论工具对服务和系统服务进行数学建模，给出模型的理论分析方法，得到服务质量性能指标的数学量化分析结果；基于性能指标的解析解，提出初步的系统优化方法，并以资源管理和任务调度两个问题为实例，给出系统优化方案；最后，基于真实数据集建立模拟实验，验证模型和分析的有效性。

3.1 服务和系统服务的性能模型

3.1.1 原子服务的性能模型

原子服务提供了最基本的服务功能，是搭建服务系统或进行服务组合的基本元素。从动态行为分析的角度来看，原子服务的动态行为主要包括如下 3 个部分。

- 1) **到达过程** 用户请求在不同时间点到达，请求原子服务提供特定的服务功能。
- 2) **排队过程** 由于服务资源有限，不能同时对所有请求进行服务，因而用户请求在到达后先进入缓冲区（即队列），等待服务可用后进行受理；若请求到达时服务空闲，则可以跳过排队过程直接进行服务。
- 3) **服务过程** 服务对用户请求进行处理，直至服务完成后离去。

上述动态过程可以由一个排队模型进行描述，如图 3.1 所示。该模型可以由一个动态事件系统（Discrete Event System, DES）进行数学描述，其中“事件”由一系列“到达”和“离去”构成，“状态”被定义为队列中请求的数目，从而该模型的状态空间为非负整数集。在时间维度上，可将时钟与状态绑定，将某个时刻 t 模型所处状态表示为 $X(t) \in \{0, 1, 2, \dots\}$ 。

$\dots\}$; 进一步, 根据请求到达的情况, 定义随机序列 $\{Y_1, Y_2, \dots\}$ 为请求到达的时间间隔, 其中 $Y_k (k \geq 2)$ 代表第 $(k-1)$ 个和第 k 个请求到达的时间间隔, Y_1 代表第一个请求的到达时间。类似地, 可以根据离去过程定义随机序列 $\{Z_1, Z_2, \dots\}$ 为服务时间, 其中 $Z_k (k \geq 1)$ 代表第 k 个请求的服务时间 (从排队结束开始服务, 到服务完成的时间间隔)。

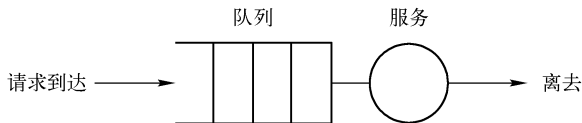


图 3.1 原子服务的排队模型

不失一般性, 假设随机序列 $\{Y_k\}$ 为独立同分布, 其概率分布函数可以由式 (3.1) 表示。该随机序列的期望值的倒数被称为**到达速率**, 形式化定义如式 (3.2) 所示。到达分布和到达速率是排队模型的重要参数。

$$A(t) = \Pr(Y \leq t) \quad (3.1)$$

$$\lambda \equiv \frac{1}{E[Y]} \quad (3.2)$$

类似地, 定义服务时间的概率分布函数如式 (3.3) 所示; 其期望值的倒数定义为**到达速率**, 如式 (3.4) 所示。

$$B(t) = \Pr(Z \leq t) \quad (3.3)$$

$$\mu \equiv \frac{1}{E[Z]} \quad (3.4)$$

除请求与到达分布参数外, 服务规程也是影响排队系统数学性质的重要参数。服务规程规定了服务可用时在队列中选择请求的顺序。对于常见的服务或服务系统, 往往采用先到先服务 (First - Come - First - Served, FCFS) 的服务策略。因此, 在后续的讨论中, 除特别指出外, 均默认采用 FCFS 的服务规程。

另外, 定义 S_k 为第 k 个请求的响应时间, 即第 k 个请求从到达时刻到服务完成离去时刻的整体经过时间。由 Little 公式可知, 若排队系统存在稳态, 则无论针对何种请求或到达分布, 式 (3.5) 恒成立。该式揭示了平均响应时间和队列状态的关系, 是性能分析的常用方法。

$$E[S] = \frac{1}{\lambda} E[X] \quad (3.5)$$

3.1.2 服务系统的性能模型

一个服务系统往往包含多个原子服务, 这些服务通过相互之间的通信和协作, 组合成为复合服务, 实现了服务系统的复杂功能。根据服务组合之间的逻辑关系, 可以将原子服务的排队模型进行串联组合, 构建排队网络模型, 从而通过相关理论和方法计算系统整体的性能指标。

一般的服务系统可以建模为一个开环的排队网络。若干个原子服务接收用户请求, 这些请求的到达构成了排队网络的外界输入弧。原子服务之间遵循一定的业务逻辑进行组合, 这些业务逻辑引发的任务流构成了排队网络内部弧, 每个内部弧连接了一个原子服务的输出和另一个原子服务的输入。有些请求需要多个原子服务协同运行, 从而在内部弧内相继扭转;

另一些则在服务完成后终结, 即该服务对应的排队模型的离去直接输出至系统外部, 该离去称为排队网络的输出弧。结合弧间关系和队列模型参数, 即可对整个排队网络进行性能分析。

以下以图 3.2 所示的一个典型的结构化服务系统为例, 介绍排队网络的性能分析方法。该系统为一个典型的层次化边缘计算架构的服务系统^[71], 服务请求可以由靠近用户的边缘层服务器处理, 也可由远端的云计算数据中心执行。同时, 边缘层可以和云端共同协作, 完成某些复杂的服务请求。

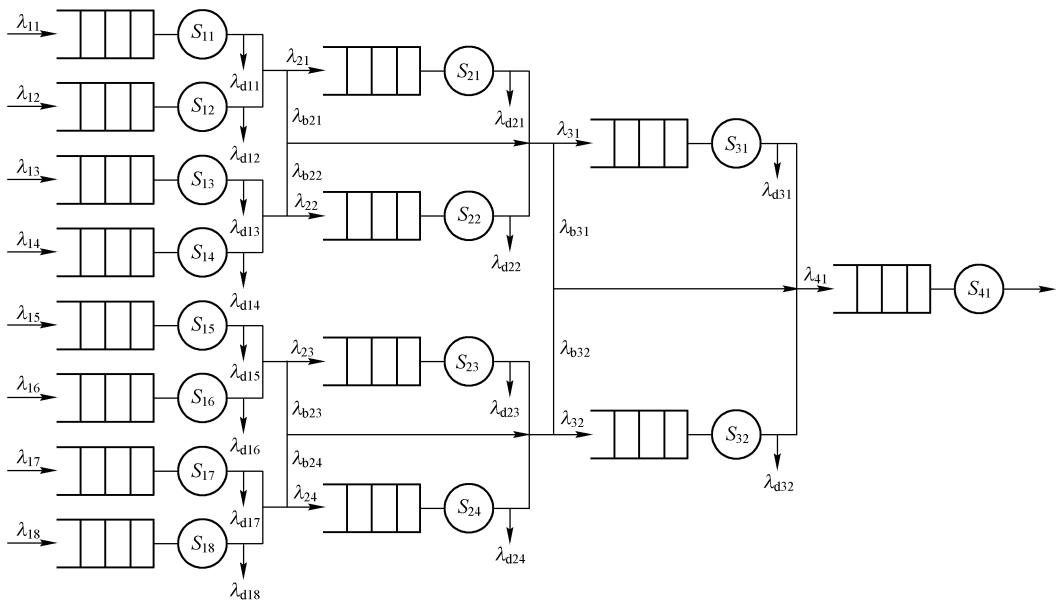


图 3.2 典型的结构化服务系统

为了便于后续的量化分析, 首先用一个二维变量 S_{ij} 对该模型中的服务器进行编号, 其中第 1 个维度 $i \in \{1, 2, \dots, l\}$ 标志了服务器所在层数, 第 2 个维度 $j \in \{1, 2, \dots, n_i\}$ 标志了该服务器在该层中的编号, 其中 n_i 代表第 i 层服务器的总数目; 其次, 设 λ_{ij} 为服务器 S_{ij} 的任务到达速率, λ_{dij} 代表服务器 S_{ij} 处理后任务离开系统的速率, λ_{bij} 代表请求在服务器 S_{ij} 处直接进行跨层向上传输而不必经过 S_{ij} 处理的请求速率。这样, 就可以通过业务逻辑计算每个服务器实际处理的任务请求速率。例如, 服务器 S_{21} 的请求到达速率为

$$\lambda_{21} = (\lambda_{11} - \lambda_{d11}) + (\lambda_{12} - \lambda_{d12}) - \lambda_{b21}$$

与排队模型类似, 用随机变量 $X_{ij} \in \{0, 1, 2, \dots\}$ 定义服务器 S_{ij} 的状态, 则通过 Little 公式, 该排队网络模型的平均响应时间的计算公式为

$$E[S] = \frac{1}{\lambda} E[X] = \frac{1}{\sum_{j=1}^{n_1} \lambda_{1j}} E \left[\sum_{i=1}^l \sum_{j=1}^{n_i} X_{ij} \right] \quad (3.6)$$

3.2 模型求解与量化分析

在 3.1 节所介绍的性能模型基础上, 可以根据实际情况对系统的性能指标进行量化分析。量化分析结果可以是数值解, 也可以是解析解。

数值解是指由模拟工具或软件通过计算机模拟技术测量得到的结果，其最常见的方法是离散事件模拟技术，即以性能模型中的到达、离去等事件作为驱动，用计算机模拟服务或服务系统的动态行为，最终通过测量、统计和计算得到性能指标结果。该模拟过程可以直接编程实现，也可以借助一些排队模型的模拟软件（如 QNet^[72]、Q-MAM^[73] 等）实现。

解析解是指通过数学计算得到的性能评价结果，它可以由数学解析式进行表达，能够清晰地揭示系统参数与性能指标之间的数量关系，对系统参数优化具有重要参考价值。但是，并非所有情况都能得到性能模型的解析解，需要在实际情况符合一定的数学规律，或者对于模型特性或参数进行一些数学假设的情况下才可以得到。本节将介绍在一定的数学规律或数学假设的前提下，如何进行性能模型的分析求解。

3.2.1 马尔可夫排队模型的求解

泊松分布到达排队模型中最常见的假设之一，其任务到达间隔时间具有指数分布性质（即无记忆性），因而被广泛应用于离散事件系统的模拟。已有相关研究指出，分布式系统会话层之上的服务请求可以近似认为服从泊松分布^[74]，对于服务计算系统中应用层的任务请求，泊松到达分布假设往往是合理的、常用的。

在本节后续的讨论中，假设原子服务的服务时间服从指数分布^[75]。该假设往往用于排队模型的最坏性能分析^[19]。直观地，其大致原因可以由如下分析得到：引入服务时间的相

关标准差变量 $c_B = \frac{\sigma_Z}{E[Z]}$ ，用以揭示服务时间这一随机变量的抖动情况。对于指数分布，有

$E[Z] = \sigma_Z = \frac{1}{\mu}$ ，即 $c_B = 1$ 。该结论表明，一个指数分布随机变量的标准差与其期望相等，

与应用层常见的其他到达分布（通常 $c_B < 1$ ）相比，其抖动情况非常剧烈，即突发情况较为明显。由排队论基本理论可知，较小的 c_B 将导致较小的平均队列长度，由 Little 公式可知，其平均响应时间也更短。因此，泊松到达假设使得性能结果比实际中的其他常见分布的更差，如果按照泊松到达模型进行系统设计，一般可以保证实际系统的性能比理论分析结果更好，因此这样的假设方法对于有 QoS 约束的优化具有很好的作用。

对于泊松到达、指数分布服务时间的原子服务排队模型，称之为 M/M/1 排队模型。在数学上，可以用一个生灭过程一维连续时间马尔可夫链（CTMC）进行形式化描述。一般地，为了保证该模型存在稳态，即队列不会无限制拥塞，假设 $\lambda < \mu$ 。定义 ρ 为服务的利用率， π_i 代表马尔可夫链状态 i 的稳态概率， q 为平均队列长度， T 为平均响应时间。根据 CTMC 的求解，结合 Little 公式，可以得到：

$$\rho \equiv \frac{\lambda}{\mu} \quad (3.7)$$

$$\pi_i \equiv \lim_{t \rightarrow \infty} \Pr(X(t) = i) = (1 - \rho)\rho^i \quad (3.8)$$

$$q \equiv E[X] = \frac{\rho}{1 - \rho} = \frac{\lambda}{\mu - \lambda} \quad (3.9)$$

$$T \equiv E[S] = \frac{1}{\mu - \lambda} \quad (3.10)$$

基于以上原子服务排队模型的分析结果，将其扩展至服务系统的排队网络模型。由

Burke 定理可知,泊松分布到达、指数分布服务时间的服务系统的离去过程也是服从泊松分布的;由 Jackson 定理可知,由该种服务系统构成的排队网络的稳定状态概率具有乘积解,即在排队网络中,每个节点可以视为一个独立的 M/M/1 排队模型。因此,可以通过式 (3.11) 计算得到整个排队网络的平均队列长度之和;进一步结合 Little 公式可以计算得到该服务系统的平均响应时间,见式 (3.12)。

$$q \equiv E[X] = E \left[\sum_{i=1}^l \sum_{j=1}^{n_i} X_{ij} \right] = \sum_{i=1}^l \sum_{j=1}^{n_i} E[X_{ij}] = \sum_{i=1}^l \sum_{j=1}^{n_i} \frac{\lambda_{ij}}{\mu_{ij} - \lambda_{ij}} \quad (3.11)$$

$$T \equiv E[S] = \frac{1}{\lambda} E[X] = \frac{\sum_{i=1}^l \sum_{j=1}^{n_i} \frac{\lambda_{ij}}{\mu_{ij} - \lambda_{ij}}}{\sum_{j=1}^{n_1} \lambda_{1j}} \quad (3.12)$$

3.2.2 半马尔可夫排队模型的求解

虽然泊松到达假设在性能评价中已被广泛认可,但为了与真实系统更加贴合,得到更为准确的评价结果,本节将对任意类型的任务到达分布进行讨论,介绍排队模型的分析方法。

首先,对原子服务的排队模型进行分析。由于请求到达可以是任意分布的,不再像泊松到达具有无记忆性,因此无法再用生灭过程马尔可夫链对队列的动态行为进行描述。但是,这里同样假设服务时间服从指数分布,该假设不仅便于最坏性能分析,也可以有助于找到一些具有无记忆性特性的时间点,即马尔可夫嵌入点。具体地,在本模型中,这些点是每个服务请求即将到达的时间点。这些马尔可夫嵌入点按照时间排序可以构成一个新的离散时间马尔可夫链(即嵌入马尔可夫链),而最初的排队模型称为半马尔可夫模型。

设 L_k 代表第 k 个请求即将到达时服务系统内的请求数目(包括排队请求和正在服务的请求), D_{k+1} 表示第 k 个和第 $k+1$ 个请求到达之间总计服务完成的数目,则有:

$$L_{k+1} = L_k + 1 - D_{k+1} \quad (3.13)$$

由于服务时间服从指数分布,具有无记忆性,因此序列 $\{L_1, L_2, \dots\}$ 构成了一个嵌入马尔可夫链,其状态 i 到状态 j 的转移概率可以由如下公式表示:

$$p_{ij} = \Pr(L_{k+1} = j | L_k = i), \quad i \geq 0, j \geq 0 \quad (3.14)$$

若 $j > i + 1$, 显然有 $p_{ij} = 0$ 。对于 $0 < j \leq i + 1$, p_{ij} 等于在相邻的两次请求到达之间正好有 $i - j + 1$ 个请求被服务完成,且服务器一直处于工作状态。对于 $j = 0$, 由于从 i 出发的所有转移概率加和为 1, p_{ij} 可以通过前两者间接计算得到。因此, p_{ij} 数学表达为

$$p_{ij} = \begin{cases} 0; & j > i + 1 \\ \int_{t=0}^{\infty} \frac{(\mu t)^{i-j+1}}{(i-j+1)!} e^{-\mu t} f_Y(t) dt; & 0 < j \leq i + 1 \\ 1 - \sum_{k=1}^{i+1} p_{ik}; & j = 0 \end{cases} \quad (3.15)$$

式中, $f_Y(\cdot)$ 是请求到达间隔时间的概率密度函数。

将 $\nu = [\nu_0, \nu_1, \dots]$ 定义为嵌入马尔可夫链的稳态概率,对该马尔可夫链进行求解,可以得到如下方程组:

$$\nu_0 = \sum_{i=1}^{\infty} \nu_i p_{i0} \quad (3.16)$$

$$\nu_n = \sum_{i=0}^{\infty} \nu_{n+i-1} \int_{t=0}^{\infty} \frac{(\mu t)^i}{i!} e^{-\mu t} f_Y(t) dt, \quad n \geq 1 \quad (3.17)$$

$$\sum_{i=1}^{\infty} \nu_i = 1 \quad (3.18)$$

对以上方程组进行数学求解, 可以发现 ν 可以表达为式 (3.19) 所示的形式, 其中 σ 是式 (3.20) 在 $0 < \sigma < 1$ 范围内的唯一解, \tilde{Y} 是请求到达间隔时间的拉普拉斯变换函数, 即

$$\tilde{Y}(s) = \int_{t=0}^{\infty} e^{-st} f_Y(t) dt。$$

$$\nu_n = (1 - \sigma) \sigma^n \quad (3.19)$$

$$\sigma = \tilde{Y}(\mu - \mu\sigma) \quad (3.20)$$

因此, 对于该半马尔可夫连, 其稳态概率的计算方法为

$$\begin{aligned} \pi_i &= \Pr(X > i - 1) - \Pr(X > i) \\ &= \rho [\Pr(L > i - 2) - \Pr(L > i - 1)] \\ &= \rho \nu_{i-1} \\ &= \rho (1 - \sigma) \sigma^{i-1} \end{aligned} \quad (3.21)$$

最后, 由 Little 公式可知, 该半马尔可夫模型对应的原子服务的响应时间为

$$T = \frac{1}{\lambda} E[X] = \frac{1}{\lambda} \sum_{i=0}^{\infty} i \cdot \pi_i = \frac{1}{\lambda} \cdot \frac{\rho}{1 - \sigma} = \frac{1}{\mu(1 - \sigma)} \quad (3.22)$$

对于半马尔可夫模型构成的排队网络, 可以对所有原子服务的平均队列长度进行统计, 结合整体的外部请求到达速率, 根据 Little 公式由式 (3.6) 计算得到。

3.3 性能优化举例

性能建模和分析对系统优化具有重要的参考价值。本节将介绍如何将上述分析结果用于系统优化, 从而保证用户的服务质量, 降低系统的部署和运行开销。本节将以资源管理和任务调度两个计算机科学领域的常见问题作为实例, 介绍基本的建模方法和优化技术。

3.3.1 服务节点的资源管理

资源管理是指将物理或虚拟服务器的计算与存储资源分配给不同的任务或用户, 从而在保证用户服务质量的前提下最大化资源利用率、最小化系统开销。多年来, 资源管理在服务计算、分布式系统、计算机网络等领域都是热门的研究课题。

考虑到资源管理的需求、形式、技术实现等都是非常复杂且多样的, 本节仅选取一个最为直观、简单的例子, 用以揭示性能模型和系统优化之间的关系。对于资源管理具体的实现细节, 本书中不做探讨。考虑一个计算资源管理的例子, 其资源分配直接影响服务器的服务能力, 因而可将资源管理问题转换为服务速率的优化问题, 优化的约束为用户规定的性能等级 (定义为响应时间的上限 T_{SLA}), 优化目标为最小化运行开销。在数学上, 该优化问题可

以表示为

$$\underset{\mu_{ij}}{\text{minimize}} \quad c(\mu_{ij}) \quad (3.23)$$

s. t.

$$T \leq T_{\text{SLA}} \quad (3.24)$$

该优化问题可以采用多种方式求解。首先,对于可以进行数学求解的情况,传统的数学优化方法或数值优化方法均可适用;其次,对于无法求得解析解,只能用模拟技术进行性能评价的情况,一些启发式方法或序优化(Ordinal Optimization)等基于搜索的方法可以保证在较高的效率下得到最优解或近似最优解。

下面讨论一个更为具体的实例。设图 3.2 所示系统中的请求到达服从泊松分布,每个节点的服务时间服从指数分布。系统共 4 层,每层服务节点同构,其请求到达速率和服务速率相同,分别为 λ_i 和 μ_i 。服务的运行开销和服务速率成正比,相关性系数为 c_0 。在常见的边缘计算系统中,第 1 层常常为数据采集或任务生成节点,很可能是传感器等硬件设备,其实现方式往往与其他节点不同,无虚拟化资源分配问题。因此,在本例中系统资源管理从第 2 层开始进行。

为了进行数学上的优化,首先对该系统的性能进行建模和分析。Burke 定理和 Jackson 定理保证了可以对每个服务节点进行独立分析,最后进行整合。因此,对于第 i 层,每个服务节点的平均队列长度为

$$q_i = \frac{\lambda_i}{\mu_i - \lambda_i} \quad (3.25)$$

由 Little 公式可知,对于整个系统服务请求的平均响应时间为

$$T = \frac{q}{\lambda} = \frac{\sum_{i=1}^4 q_i n_i}{\lambda_1 n_1} = \frac{\frac{\lambda_1 n_1}{\mu_1 - \lambda_1} + \frac{\lambda_2 n_2}{\mu_2 - \lambda_2} + \frac{\lambda_3 n_3}{\mu_3 - \lambda_3} + \frac{\lambda_4 n_4}{\mu_4 - \lambda_4}}{\lambda_1 n_1} \quad (3.26)$$

因此,优化问题的建模表达可以更加具体化,即

$$\underset{\mu_2, \mu_3, \mu_4}{\text{minimize}} \quad c(\mu_2, \mu_3, \mu_4) = c_0(n_2 \mu_2 + n_3 \mu_3 + n_4 \mu_4) \quad (3.27)$$

s. t.

$$\frac{\lambda_1 n_1}{\mu_1 - \lambda_1} + \frac{\lambda_2 n_2}{\mu_2 - \lambda_2} + \frac{\lambda_3 n_3}{\mu_3 - \lambda_3} + \frac{\lambda_4 n_4}{\mu_4 - \lambda_4} \leq T_{\text{SLA}} \quad (3.28)$$

上述问题是一个典型的非线性优化问题。为了对其进行求解,首先将式(3.28)转化为如下形式:

$$g(\mu_2, \mu_3, \mu_4) = \lambda_1 n_1 T_{\text{SLA}} - \left(\frac{\lambda_1 n_1}{\mu_1 - \lambda_1} + \frac{\lambda_2 n_2}{\mu_2 - \lambda_2} + \frac{\lambda_3 n_3}{\mu_3 - \lambda_3} + \frac{\lambda_4 n_4}{\mu_4 - \lambda_4} \right) \geq 0 \quad (3.29)$$

随后,引入一个非负参数 $\theta \geq 0$ 。由 KKT (Karush - Kuhn - Tucker) 条件可知,最优解存在于如下方程:

$$\frac{\partial c(\mu_2, \mu_3, \mu_4)}{\partial \mu_2} - \theta \frac{\partial g(\mu_2, \mu_3, \mu_4)}{\partial \mu_2} = c_0 n_2 - \theta \frac{\lambda_2 n_2}{(\mu_2 - \lambda_2)^2} = 0 \quad (3.30)$$

$$\frac{\partial c(\mu_2, \mu_3, \mu_4)}{\partial \mu_3} - \theta \frac{\partial g(\mu_2, \mu_3, \mu_4)}{\partial \mu_3} = c_0 n_3 - \theta \frac{\lambda_3 n_3}{(\mu_3 - \lambda_3)^2} = 0 \quad (3.31)$$

$$\frac{\partial c(\mu_2, \mu_3, \mu_4)}{\partial \mu_4} - \theta \frac{\partial g(\mu_2, \mu_3, \mu_4)}{\partial \mu_4} = c_0 n_4 - \theta \frac{\lambda_4 n_4}{(\mu_4 - \lambda_4)^2} = 0 \quad (3.32)$$

$$\theta \cdot g(\mu_2, \mu_3, \mu_4) = 0 \quad (3.33)$$

$$\theta \geq 0 \quad (3.34)$$

考虑到本身决策变量的定义域，得到该优化问题的最优解为

$$\mu_2 = \frac{n_2 \lambda_2 + n_3 \sqrt{\lambda_2 \lambda_3} + n_4 \sqrt{\lambda_2 \lambda_4}}{\lambda_1 n_1 \left(T_{\text{SLA}} - \frac{1}{\mu_1 - \lambda_1} \right)} + \lambda_2 \quad (3.35)$$

$$\mu_3 = \frac{n_2 \sqrt{\lambda_2 \lambda_3} + n_3 \lambda_3 + n_4 \sqrt{\lambda_3 \lambda_4}}{\lambda_1 n_1 \left(T_{\text{SLA}} - \frac{1}{\mu_1 - \lambda_1} \right)} + \lambda_3 \quad (3.36)$$

$$\mu_4 = \frac{n_2 \sqrt{\lambda_2 \lambda_4} + n_3 \sqrt{\lambda_3 \lambda_4} + n_4 \lambda_4}{\lambda_1 n_1 \left(T_{\text{SLA}} - \frac{1}{\mu_1 - \lambda_1} \right)} + \lambda_4 \quad (3.37)$$

3.3.2 分层系统的任务调度

任务调度旨在设计有效的策略，用以将任务请求分配给不同的服务或服务器，从而在满足用户性能需求的前提下最大化服务收益。为了简单起见，将任务调度问题规约为服务节点间请求速率的分配问题。

与 3.3.1 节类似，可以将任务调度建模为一个有约束的非线性优化问题。优化目标仍为最小化服务运行成本，优化变量为每个服务节点的请求速率。在本问题中，约束分为两个部分，一是用户的性能要求必须被满足，二是所有的请求必须被处理。具体地，该优化问题的模型为

$$\underset{\lambda_{ij}, i > 1}{\text{minimize}} \quad c(\lambda_{ij}) \quad (3.38)$$

s. t.

$$T \leq T_{\text{SLA}} \quad (3.39)$$

$$\sum_{i=2}^l \sum_{j=1}^{n_i} \lambda_{ij} \geq \sum_{j=1}^{n_1} \lambda_{1j} \quad (3.40)$$

同样，讨论图 3.2 所示的泊松到达指数服务时间的层次化服务系统。该系统的各层内部服务节点同构，服务的运行开销正比于所服务的任务数，数学上可以理解为其正比于服务请求的到达速率，比例参数设为 c_i 。类似 3.3.1 节所述的分析，可知式 (3.25) 和式 (3.26) 在本例中同样成立。因此，优化问题可以具体化为

$$\underset{\lambda_2, \lambda_3, \lambda_4}{\text{minimize}} \quad c(\lambda_2, \lambda_3, \lambda_4) = c_2 n_2 \lambda_2 + c_3 n_3 \lambda_3 + c_4 n_4 \lambda_4 \quad (3.41)$$

s. t.

$$\frac{\lambda_1 n_1}{\mu_1 - \lambda_1} + \frac{\lambda_2 n_2}{\mu_2 - \lambda_2} + \frac{\lambda_3 n_3}{\mu_3 - \lambda_3} + \frac{\lambda_4 n_4}{\mu_4 - \lambda_4} \leq T_{\text{SLA}} \quad (3.42)$$

$$n_2 \lambda_2 + n_3 \lambda_3 + n_4 \lambda_4 \geq n_1 \lambda_1 \quad (3.43)$$

式 (3.42) 和式 (3.43) 所述的约束条件可以转化为更标准的表达形式，即

$$g(\lambda_2, \lambda_3, \lambda_4) = \lambda_1 n_1 T_{SLA} - \left(\frac{\lambda_1 n_1}{\mu_1 - \lambda_1} + \frac{\lambda_2 n_2}{\mu_2 - \lambda_2} + \frac{\lambda_3 n_3}{\mu_3 - \lambda_3} + \frac{\lambda_4 n_4}{\mu_4 - \lambda_4} \right) \geq 0 \quad (3.44)$$

$$h(\lambda_2, \lambda_3, \lambda_4) = n_2 \lambda_2 + n_3 \lambda_3 + n_4 \lambda_4 - n_1 \lambda_1 \geq 0 \quad (3.45)$$

同样, 依据 KKT 条件, 最优解可以通过求解如下方程组得到:

$$\frac{\partial c(\lambda_2, \lambda_3, \lambda_4)}{\partial \lambda_2} - \theta_1 \frac{\partial g(\lambda_2, \lambda_3, \lambda_4)}{\partial \lambda_2} - \theta_2 \frac{\partial h(\lambda_2, \lambda_3, \lambda_4)}{\partial \lambda_2} = 0 \quad (3.46)$$

$$\frac{\partial c(\lambda_2, \lambda_3, \lambda_4)}{\partial \lambda_3} - \theta_1 \frac{\partial g(\lambda_2, \lambda_3, \lambda_4)}{\partial \lambda_3} - \theta_2 \frac{\partial h(\lambda_2, \lambda_3, \lambda_4)}{\partial \lambda_3} = 0 \quad (3.47)$$

$$\frac{\partial c(\lambda_2, \lambda_3, \lambda_4)}{\partial \lambda_4} - \theta_1 \frac{\partial g(\lambda_2, \lambda_3, \lambda_4)}{\partial \lambda_4} - \theta_2 \frac{\partial h(\lambda_2, \lambda_3, \lambda_4)}{\partial \lambda_4} = 0 \quad (3.48)$$

$$\theta_1 \cdot g(\lambda_2, \lambda_3, \lambda_4) = 0 \quad (3.49)$$

$$\theta_2 \cdot h(\lambda_2, \lambda_3, \lambda_4) = 0 \quad (3.50)$$

$$\theta_1 \geq 0, \theta_2 \geq 0 \quad (3.51)$$

由于该方程的解析解过于复杂, 在此不便于全部写出。除以上的数学解法外, 本节所述的优化问题还可以通过其他方法得到。例如, 可以利用数值分析方法得到最优或局部最优解, 如牛顿—拉夫森 (Newton – Raphson) 迭代法、拟牛顿法 (Quasi – Newton)、共轭梯度法等。另外, 可以采用启发式算法进行快速搜索, 如模拟退火算法、遗传算法、蚁群算法等。具体使用何种算法, 可以根据优化问题的复杂程度、最优解的分布情况、算法的效率要求等实际情况来确定。

3.4 案例分析

本节以真实的开源数据集为基础进行模拟实验, 采用离散事件模拟技术对服务系统进行仿真, 对结果进行分析, 用以验证评价结果。

3.4.1 数据集介绍

本节选用微软亚洲研究院公开的“T – Drive”数据集^[76,77]作为模拟实验的基础。该数据集记录了2008年北京市共10357辆出租车历时一周的GPS数据, 共计约1500万条数据, 覆盖超过 9×10^6 km的路线距离。在该数据集中, 不仅记录了这些出租车的经度、纬度等位置信息, 还在每条数据中清晰地记录了数据上传的时间。这些时间信息反映了该数据采集和处理服务系统的请求到达情况, 是性能建模的重要参数。分布在各地的出租车上的传感器进行数据采集, 然后将其上传至分布式服务系统中进行汇总和计算处理。这些传感器和分布式系统构成了一个典型的以边缘计算为模式的物联网服务系统。

对数据集进行深入分析后可以发现, 不同出租车的GPS传感器具有不同的采样周期和分布, 绝大多数请求的平均间隔时间在1~20min范围。图3.3所示的是两种典型的请求到达分布。图3.3(a)所示的是18号出租车的传感器请求到达时间间隔的累积密度函数, 呈现出“阶梯状”规律, 即多数的请求时间间隔具有相同的数值。具体地, 从数据分析图中可知, 约90%的请求间隔为300s, 即该传感器基本上固定每5min向系统上报一次GPS信息; 5%的数据数值为0, 即可能由于信息冗余或重传同一时刻上报了两次数据; 有4%的间

隔时间为 10min，该情况有可能是由于其中有一次传输丢包导致。图 3.3 (b) 所示的是另一种典型的请求到达分布情况。对于此类请求，间隔时间在 5min 之内的数据类似于随机均匀分布，仍有绝大多数请求固定在 5min 的时间间隔。另外，超过 95% 的数据都在 10min 以内。对于单一服务请求的到达，可以大致通过确定分布进行近似分析。

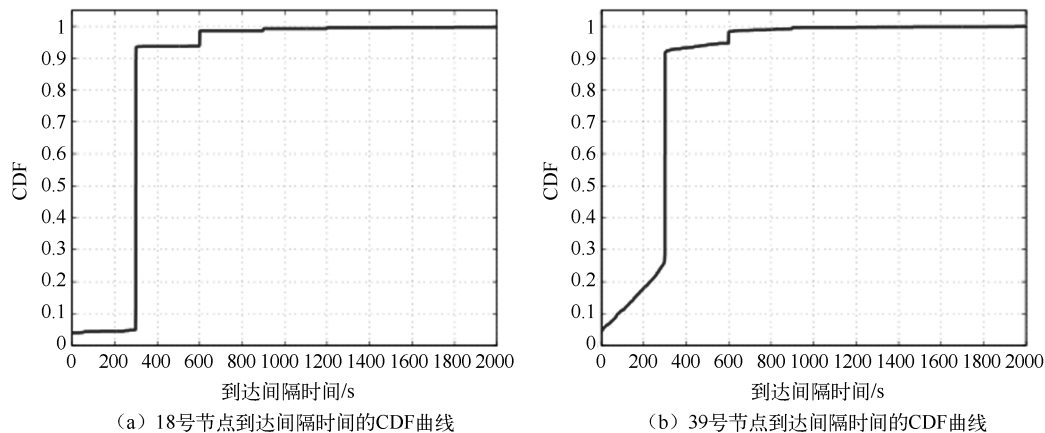


图 3.3 请求时间间隔的分布函数

进一步，随机选取了若干个出租车，将其请求进行聚集，模拟服务系统中存在请求聚集器或传感器集线器的情况，进而对聚集服务请求的到达分布进行分析讨论。图 3.4 所示的是聚集服务请求时间间隔的分布函数。通过近似分析可以发现，聚集服务请求的时间间隔大致服从指数分布，即聚集服务请求可以用泊松分布进行近似建模。该数据分析结果验证了前述章节泊松到达假设的合理性。

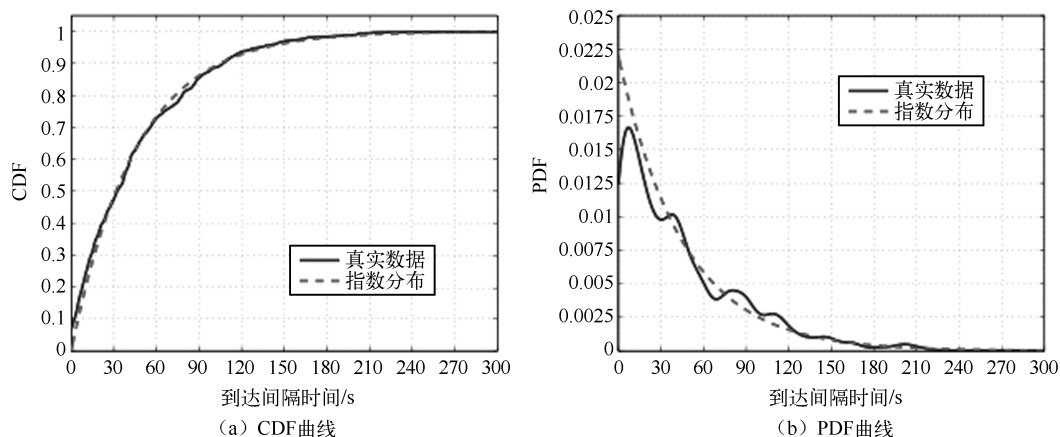


图 3.4 聚集服务请求时间间隔的分布函数

3.4.2 原子服务的模拟实验

为了验证模型的有效性，首先构建原子服务的模拟程序。所有的服务请求根据 T - Drive 真实数据集的记录进行生成，原子服务的服务时间根据期望为 150s 的指数分布进行随机生成。模拟程序描述了一个随机排队系统，事件样本根据随机过程和排队动态行为进行生成。

模拟过程中的统计数据被悉数记录，用以进行系统性能分析，进而与理论分析结果进行比较，对理论模型进行校验。实验中，分别对确定到达分布和泊松到达分布两种假设进行理论分析。由前述章节讨论可知，确定到达分布更符合单一数据源的实际情况，而泊松到达分布假设的模型（马尔可夫排队模型）可以模拟聚集请求到达情形，或者可以给出最坏性能分析结果。

原子服务的模拟实验结果如图 3.5 所示。实验中，请求到达由单一的出租车 GPS 传感器生成。从实验结果可见，确定到达分布假设下的理论分析结果与模拟实验得到的统计结果非常相近，误差可控制在 4% 以内，因而该模型可以较好地应用于此类物联网服务的性能评价。另一方面，马尔可夫排队模型给出的性能分析结果与实验结果相差较大，甚至误差可能超过 50%。对于此类模型，请求到达的突发性比确定性到达显著提高，排队等待更易发生，因而平均响应时间也更大。因此，泊松到达模型可以模拟突发性较大的请求到达情况，对一般系统而言给出的是最坏性能分析预测。

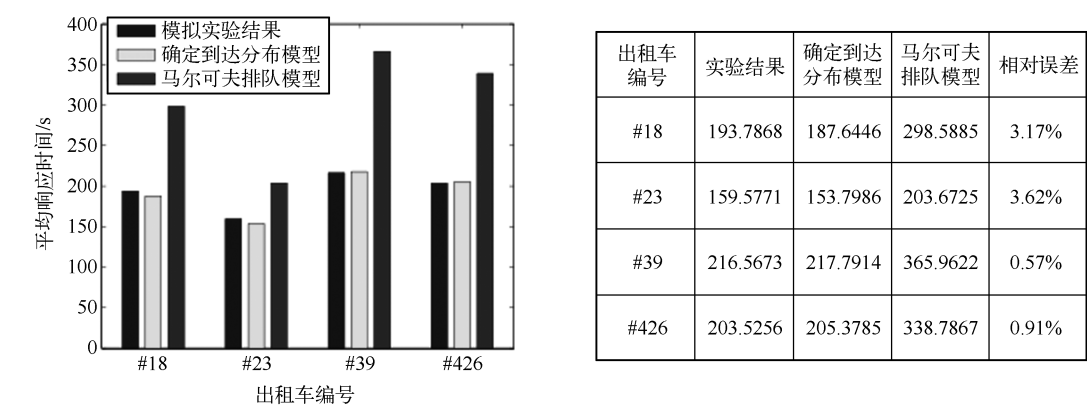


图 3.5 原子服务模拟实验结果

此外，可以将多个出租车的传感器数据聚集，模拟聚集请求到达情形下的原子服务。同样，分别采用马尔可夫和半马尔可夫两种排队模型进行理论分析，与模拟实验进行比对。图 3.6 所示的是不同服务速率情况下两种理论模型相对于模拟实验结果的相对误差。

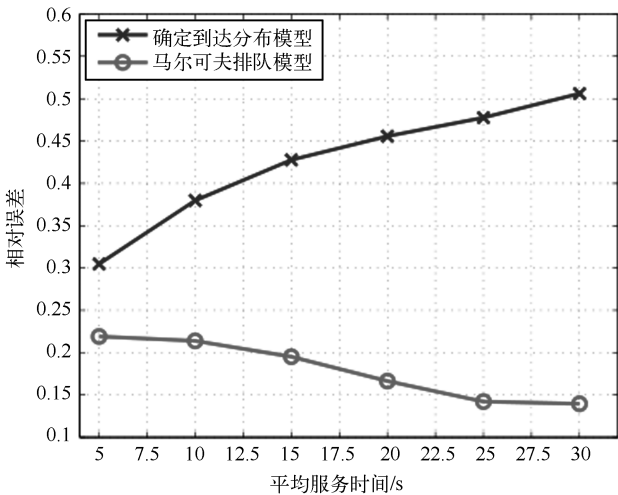


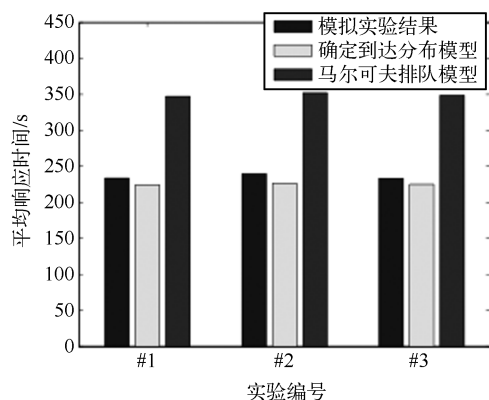
图 3.6 聚集请求到达模拟实验结果

由图可见，马尔可夫模型对性能预测的结果相对更加准确，相对误差可以控制在 25% 以内，并且随着平均服务时间的增大准确性逐渐增加。上述原因在于：相对于确定时间到达，泊松到达更符合实际数据情况，因而模型分析结果与实验结果更为接近；平均服务时间的增大会导致更大概率的拥塞发生，造成系统性能的下降，因而与最坏性能分析的马尔可夫排队模型更为贴近。因此可以得出的结论是，对于聚集请求到达之类的突发性较强的请求到达情形，马尔可夫模型可以得到相对更为准确的性能分析结果。

3.4.3 服务系统的模拟实验

为了验证排队模型的有效性，将原子服务进行组合，构建一个服务系统的模拟实验。在实验中，原子服务通过层次式结构关系进行组合，如图 3.2 所示。每个 GPS 传感器将服务请求直接上传至边缘节点服务器，进而根据需要由上层服务进行计算和处理。

首先，从 T-Drive 数据集随机选取不同的服务到达请求作为实验的输入参数，运行模拟实验，对结果进行分析。图 3.7 所示的是服务系统不同请求到达的模拟实验结果。可见，确定到达分布的半马尔可夫模型的理论结果与模拟实验结果相近，误差范围为 3~6%，而马尔可夫排队模型更适用于最坏性能分析。



实验编号	实验结果	确定到达分布模型	马尔可夫排队模型	相对误差
#1	233.8625	224.4584	347.4407	4.02%
#2	239.8189	226.5554	352.1432	5.53%
#3	233.6540	225.1433	349.0132	3.64%

图 3.7 服务系统不同请求到达的模拟实验结果

另一方面，在同样的请求到达情况下，调整系统参数，讨论参数与性能结果之间的关系，进而进行系统优化。首先，分别调整第 2 层和第 4 层的服务速率，如图 3.8 (a) 所示。从图中可见，任一层服务速率的增大都可以带来整个系统性能的提升，但是在此参数设置的情形下，第 2 层（即边缘服务节点）是性能瓶颈，其服务速率的提高可以为系统性能提升带来非常显著的效果。因此，在有限的资源下，提高边缘服务节点的计算能力相比于云端节点可以为服务供应商带来更多的性能回报。其次，调整任务请求的分配比率，模拟不同任务调度策略下的服务系统动态行为。图 3.8 所示的是调整任务请求的分配比率的模拟实验结果。由于云端服务节点的计算能力更强，服务速率更快，因而增大跨层概率（即分配更多任务予云端服务节点）可以带来性能提升。此外，提高边缘层任务的跨层比率相比于云层可以带来明显的性能提升，因而在此类层次结构的服务系统中，边缘层和云端之间的任务调度往往比云计算系统内部任务调度更加重要，需要对此引起更多的关注和研究。

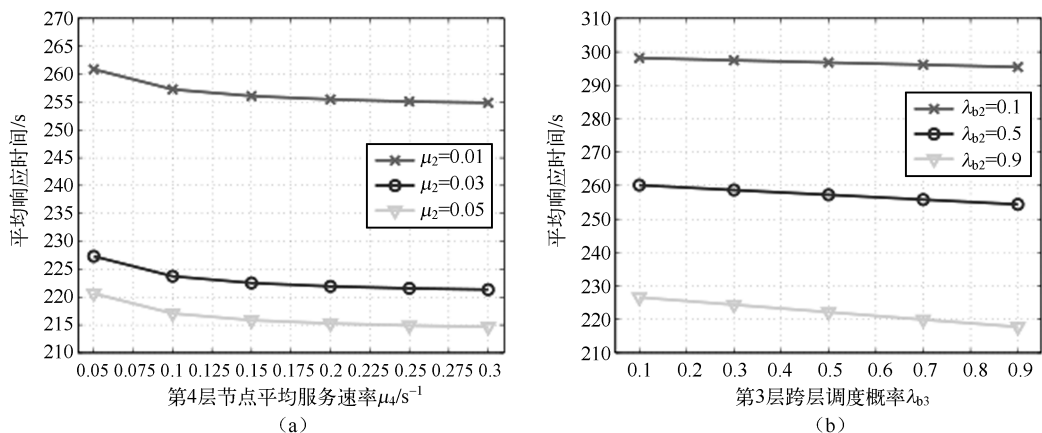


图 3.8 服务系统不同参数的模拟实验结果

3.5 本章小结

性能是服务质量首要关注的指标，对于服务质量的保障和优化具有重要意义。本章针对不同情形的请求到达，建立了服务和服务系统动态行为的数学模型，并基于随机过程分析给出了原子服务和服务系统性能的量化评价。本章以服务节点的资源管理和层次式服务系统的任务调度为例，介绍了基于随机模型量化分析的服务质量优化方法。最后，基于真实数据集构建模拟实验，验证了随机建模的有效性和量化分析的正确性。

基于半马尔可夫模型的可信赖性评价

随着服务计算应用领域的不断扩大,可信赖性受到了越来越多的关注,业已成为服务的一个关键需求。服务可信赖性的分析、评价,将对服务质量的描述和保障,服务之间的比较和选择,以及服务需求的确定提供理论依据。

传统的可信赖性分析方法往往从系统或组件的角度展开建模和分析。而在服务计算中,由于松耦合、动态性和大规模的特点,这类方法可能不再适用或有效。具体地,服务计算中松耦合特性使得上层服务和底层硬件不再具有固定的耦合对应关系,自下而上的分析方法无法直观反映服务的特性,需要对复杂的软/硬件关系进行研究,限制了建模和分析方法的效率。在服务的动态性要求评价中,需要对服务的选择、组合和协同提供理论支持,这一点在传统的可信赖性建模中往往被忽视,需要在理论上进行扩充。服务的大规模特性要求高效的建模方法和求解技术,需要在传统模型方法中引入状态聚合等技术。

针对这些难点,已有部分学者进行了一些初探。一方面,部分研究从用户或服务的角度出发,研究可信赖性相关的属性,将用户视为一个动态系统,从用户的角度进行系统建模,对可靠性、可用性等属性进行了分析^[57,78,79]。另一方面,一些研究从某些特定系统的特性出发,有针对性地进行可信赖性分析评价,如 SOA 系统^[80]、集中管理的异构分布式系统^[81]、网络服务系统^[82]等。虽然这些研究对服务计算的可信赖性评价具有一定的推动作用,但仍存在不足。首先,已有工作大部分只讨论可靠性或可用性等部分属性,缺乏对可信赖性的全面系统性研究;其次,部分研究针对特定系统,限制了模型的普适性,目前尚无对服务计算系统整体的抽象模型;此外,目前的研究往往忽视对可信赖性属性之间的相互关系的探讨,理论体系不够全面。

对此,本章将从面向服务的角度,对服务计算系统的可信赖性进行全面、系统的讨论;从服务状态出发,建立服务计算系统的状态转移模型,给出可信赖属性的数学量化表达;进一步,研究可信赖性维度内部分属性之间的相互关系,给出它们的数学表示;另外,对服务计算中服务组合的基本模式进行讨论,给出服务组合的可信赖性分析方法。

4.1 服务计算系统的可信赖性建模

4.1.1 基于半马尔可夫过程的系统状态建模

在服务计算系统的运行过程中,可能会遇到错误、故障、攻击、修复等一系列复杂的事

件, 这些事件将影响系统的运行状态, 从而决定了系统接受、处理和完成服务的能力。根据服务计算系统中可能遇到的与可信赖性相关的事件, 讨论系统的状态转移, 分析系统状态的动态变化, 对于可信赖性评价具有重要意义。

文献 [57] 对分布式系统的错误和故障状态进行了初探性的总结, 从用户的角度出发, 建立了系统的状态转移模型。但是, 该模型仅针对可靠性和可用性两个属性, 虽然根据可信赖性属性的定义, 该模型在一定情况下可以用于可维护性和保险性的评价, 但完整性评价始终无法完成。因此, 本书对该模型进行了修改和补充, 完善了系统状态和相关事件, 使其可以对可信赖性的所有属性进行评价。

首先, 从面向服务的角度建立服务计算系统的状态转移图, 如图 4.1 所示。系统状态可分为可运行 (Ready) 和故障 (Failed) 两类, 而系统的完整性保证与否将可运行状态细分为完整性保证 (Integrity – Maintained) 和完整性破坏 (Integrity – Destroyed, ID) 两类。其中, 完整性保证前提下的可访问 (Accessible, A) 状态为系统中与系统周边的所有部件均无错误或故障的正常状态。在系统的运行过程中, 虽然系统所有部件均正常运行, 但可能由于周边网络故障、网络能力有限或遭受拒绝服务 (Denial of Service, DoS) 攻击, 用户的服务请求无法到达服务计算系统, 从而无法完成所需的服务要求, 本书将这样的状态定义为不可访问 (Inaccessible, IA) 状态。此外, 系统在运行过程中还可能由于软/硬件故障、人为管理失误或外界攻击使系统遭受不恰当的部件、模块或数据的更改, 如果该更改尚未造成系统故障或服务失效, 则定义这样的系统状态为完整性破坏 (Integrity – Destroyed, ID) 状态。

在服务计算系统的运行过程中, 可能会遇到各种错误或故障, 如人为操作失误、软/硬件

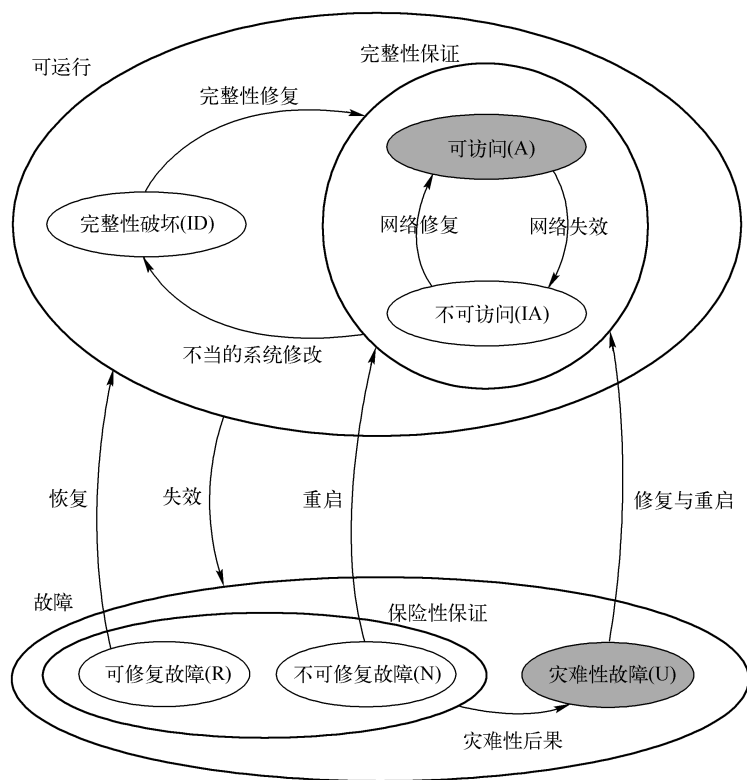


图 4.1 服务计算系统的状态转移图

缺陷、恶意攻击、病毒、断电、硬件故障等。在某些情况下，一些错误或故障可以被避免或容忍，如虚拟机动态迁移机制可以将承载服务的虚拟机从故障物理机移至另一台物理机，从而避免服务失效^[11]。而另一方面，如果不能被避免或容忍，则服务系统将进入故障（Failed）状态。根据故障的严重程度，也可将其细分为可修复故障（Recoverable, R）、不可修复故障（Nonrecoverable, N）和灾难性故障（Unsafe, U）。可修复故障代表系统可以通过自动恢复机制修复已有故障并回到可运行状态；而不可修复故障则必须经过外界干预并进行系统重启后，才能恢复系统正常运行的状态^[57]；灾难性故障代表该故障可能对用户或周围环境造成灾难性（Catastrophic）的后果。这里，对“灾难性”的定义可以由用户或服务提供商决定。例如，汽车系统中中央控制行车系统的故障可以被定义为“灾难性的”，该系统的失效可能造成严重的安全事故^[83]；在大规模数据中心中，冷却系统故障可能造成大规模硬件损坏，甚至导致火灾，该故障在这类服务计算系统中可以被视为灾难性故障。

从可信赖性和安全性评价的角度出发，有些状态逗留时间的分布可能是非指数分布^[39]。例如，文献[59]通过大量的实际数据分析得到，两次故障之间的间隔时间（即正常状态的逗留时间）呈现出类似伽马分布（Gamma Distribution）或韦伯分布（Weibull Distribution）。因此，为了更加准确地描述系统的动态行为，采用半马尔可夫过程（SMP）对系统状态转移进行建模。与可信赖性评价常用的 CTMC 模型相比，该模型更具有普适性。显然，如果状态逗留时间服从指数分布，该 SMP 模型也可准确地对该情况进行描述，CTMC 是 SMP 的一个特例，SMP 具有更为广泛的应用范围。

4.1.2 可信赖性形式化定义

文献[14]从宏观上给出了可信赖性的各个属性的定义和说明，该模型旨在全面覆盖所有计算机和通信系统，给出广义且普适的可信赖性理论框架，因而忽略了对具体系统的讨论，从而未能给出每个属性的数学表达。本书从普遍公认的可信赖性属性的定义出发，结合服务计算系统的状态转移模型，给出各个属性的形式化定义，以及量化计算方法。

1) 可用性 可用性（Availability）代表了当用户需要时系统可以向用户提供正常工作的服务的能力。在 SMP 模型中，某个时刻 t 的系统可用性可以用当前时刻系统处于可访问状态的概率表示，即

$$A(t) = \Pr(\text{Accessible}(t)) \quad (4.1)$$

2) 完整性 完整性（Integrity）反映了服务计算系统抵御不适当的系统或数据修改的能力。服务过程中，一般用户更关注于服务运行时关键的组件或数据是否得到保护，因此完整性可以用完整性保证状态在可运行状态下的条件概率来表达，如：

$$I(t) = \Pr(\text{Integrity} - \text{maintained}(t) \mid \text{Ready}(t)) \quad (4.2)$$

3) 保险性 保险性（Safety）是服务在其生命周期内不会对用户和环境造成灾难性后果的能力。它反映了系统发生故障后避免造成灾难性后果的能力，因而可以通过式（4.3）所示的条件概率进行定义和评价。

$$S(t) = \Pr(\text{Safe}(t) \mid \text{Failed}(t)) \quad (4.3)$$

4) 可维护性 可维护性（Maintainability）描述了系统可以进行调整、修复和容错的能力。它可以通过系统处于可修复故障状态在所有故障状态中的条件概率得以评价，即

$$M(t) = \Pr(\text{Recoverable}(t) \mid \text{Failed}(t)) \quad (4.4)$$

5) 可靠性 可靠性 (Reliability) 是指服务可以在一个特定的时间区间内持续正确工作的概率。假设在时刻 t 系统处于可运行状态, 则可靠性可以由 t 至 $t + \tau$ 时刻持续处于可运行状态的概率表示^[84], 见式 (4.5)。式中, $\text{Ready}(t, t + \tau)$ 表示系统在 $[t, t + \tau]$ 时间段内均处于可运行状态。

$$R(t, \tau) = \Pr(\text{Ready}(t, t + \tau) \mid \text{Ready}(t)) \quad (4.5)$$

4.2 模型分析与可信赖性评价

4.2.1 稳态概率分析

为了分析可信赖性属性, 首先需要对 SMP 状态转移模型进行求解。假设已知: (1) 各个状态的平均逗留时间 h_i , 其中 $i \in \{A, IA, ID, R, N, U\}$; (2) 各个状态之间的转移概率矩阵 \mathbf{P} , 见式 (4.6)。本节关注于介绍服务计算系统可信赖性的建模和分析方法, 因而不在此处详细讨论模型中的参数测量和参数估计的问题。在 4.5 节中, 将展示一种基于统计的从系统日志文件获取模型参数的方法, 用以在实际系统中应用本章所提出的模型进行可信赖性评价分析。

$$\mathbf{P} = \begin{matrix} & \begin{matrix} A \\ IA \\ ID \\ R \\ N \\ U \end{matrix} \end{matrix} \begin{bmatrix} 0 & p_{IA} & p_{ID} & p_R & p_N & p_U \\ p_A & 0 & p_{ID} & p_R & p_N & p_U \\ p_{IDA} & p_{IDIA} & 0 & p_R & p_N & p_U \\ p_{RA} & p_{RIA} & p_{RID} & 0 & 0 & p_{RU} \\ p_{NA} & p_{NIA} & 0 & 0 & 0 & p_{NU} \\ p_{UA} & p_{UIA} & 0 & 0 & 0 & 0 \end{bmatrix} \quad (4.6)$$

为了求解 SMP 模型, 首先选取状态发生变化的时间点作为嵌入点, 将 SMP 嵌入一个离散时间马尔可夫链 (DTMC) 中。该嵌入 DTMC 的转移概率矩阵如式 (4.6) 所示。另外, 考虑到任意状态到其他所有状态的转移概率之和应为 1, 因而转移概率参数应满足如下归一化条件:

$$\begin{aligned} p_{IA} + p_{ID} + p_R + p_N + p_U &= 1 \\ p_A + p_{ID} + p_R + p_N + p_U &= 1 \\ p_{IDA} + p_{IDIA} + p_R + p_N + p_U &= 1 \\ p_{RA} + p_{RIA} + p_{RID} + p_{RU} &= 1 \\ p_{NA} + p_{NIA} + p_{NU} &= 1 \\ p_{UA} + p_{UIA} &= 1 \end{aligned}$$

从转移概率矩阵 \mathbf{P} 的数学表达可知, 该嵌入 DTMC 是不可约 (Irreducible)、非周期的 (Aperiodic), 因而其稳态分布必然存在, 且与初始状态无关。通过求解式 (4.7) 所示的线性方程组, 可以得到嵌入 DTMC 各状态的稳态概率向量 $\boldsymbol{\nu} = [\nu_A, \nu_{IA}, \nu_{ID}, \nu_R, \nu_N, \nu_U]$ 。

$$\begin{cases} \nu \cdot P = \nu \\ \sum_i \nu_i = 1 \end{cases} \quad (4.7)$$

继而, 由嵌入 DTMC 的稳态概率 ν , 以及 SMP 模型中每个状态的平均逗留时间 h_i , 可以由式 (4.8) 计算得到 SMP 模型中任意状态 $i \in \{A, IA, ID, R, N, U\}$ 的稳态概率 π_i , 即

$$\pi_i = \frac{\nu_i h_i}{\sum_j \nu_j h_j}, \quad i \in \{A, IA, ID, R, N, U\} \quad (4.8)$$

从系统设计和优化的角度出发, 一般来讲, 稳态的可信赖性分析比瞬态分析 (即分析某个时刻 t 系统的可信赖性属性) 更有意义。在已有的服务计算系统 SMP 模型稳态概率分析的基础上, 根据可信赖性的形式化定义, 可以得到如下的可信赖性属性的稳态量化表达。

1) **可用性** 根据式 (4.1) 的定义, 系统的可用性可以由 SMP 模型中可访问状态的稳态概率来表示, 如:

$$A = \Pr_S(A) = \lim_{T \rightarrow \infty} \frac{1}{T} \int_0^T \Pr(\text{Accessible}(t)) = \pi_A \quad (4.9)$$

2) **完整性** 由式 (4.2), 稳态的完整性表达可以表示为如下条件概率:

$$I = \Pr_S(\text{Integrity} - \text{maintained} | \text{Ready}) = \frac{\pi_A + \pi_{IA}}{\pi_A + \pi_{IA} + \pi_{ID}} \quad (4.10)$$

3) **保险性** 式 (4.3) 所示的条件概率的稳态值可用来评价服务计算系统的保险性, 即

$$S = \Pr_S(\text{Safe} | \text{Failed}) = \frac{\pi_R + \pi_N}{\pi_R + \pi_N + \pi_U} \quad (4.11)$$

4) **可维护性** 可维护性可以通过系统处于可修复故障状态在所有故障状态中的条件概率稳态值得以评价, 即

$$M = \Pr_S(\text{Recoverable} | \text{Failed}) = \frac{\pi_R}{\pi_R + \pi_N + \pi_U} \quad (4.12)$$

5) **可靠性** 在稳态分析中, 可以假设系统在时刻 0 处于可运行状态, 则稳态可靠性可以由式 (4.13) 评价。另外, 可靠性也可以通过平均失效时间 (MTTF) 加以评价, 其具体的评价方法和量化分析将在 4.2.2 节给出。

$$R(\tau) = \Pr(\text{Ready}(0, \tau) | \text{Ready}(0)) \quad (4.13)$$

4.2.2 平均失效时间分析

平均失效时间 (MTTF) 是指系统从正常状态开始, 直至转移到某个或某些人为定义的故障状态需要的平均时间, 是可靠性分析中的一个常用的评价指标^[39]。本书扩充了 MTTF 的概念, 给出不同属性的 MTTF 分析。从可信赖性属性的定义出发, 针对不同属性的评价过程, 可以分别定义不同的正常状态和故障状态, 从而得到不同属性的 MTTF 值。这样, 除前述的稳态概率分析外, 可以从时间维度这一不同的角度对可信赖性属性进行评价。

在模型分析中, 将系统的正常状态和故障状态分别定义为瞬态 (Transient State) 和吸收态 (Absorbing State), 这两类状态的区分方法由待评价的可信赖性属性的定义决定。模型中, 一旦系统进入吸收态状态, 则从该状态转出的概率为 0。这样, 从某个初始状态 q (正

常状态、瞬态) 出发, 分析每个瞬态的平均访问次数, 结合其逗留时间, 即可计算得到 MTTF 的量化表达^[46]。

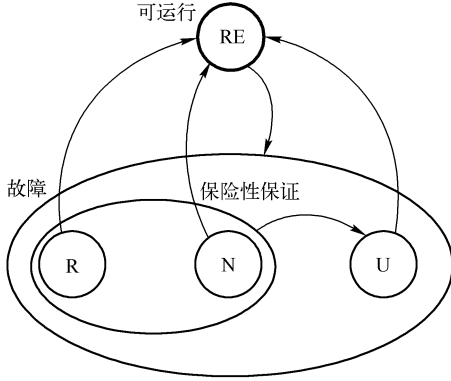


图 4.2 简化的服务计算系统状态转移图

这里, 为了进一步降低分析的复杂度, 直观显示评价结果和系统特性, 将可运行状态中的所有子状态进行状态聚合, 提出了一种简化的系统状态转移 SMP 模型, 如图 4.2 所示。模型的简化便于可靠性、保险性和可维护性的分析。这些属性和可运行状态中的子状态关联度较小, 分析时不必考虑可运行状态中的状态转移情况, 因而在不影响模型分析准确程度的前提下, 简化分析过程和结果的表达。此外, 在实际系统中, 有些状态的测量或记录比较困难, 如在一些系统故障日志中可能没有完整性被破坏的相关数据记

载^[85]。因此, 该简化模型可以在系统参数测量不全面的情况下, 对其他可信赖性属性进行评价, 具有更好的适用范围。

该简化模型中, 嵌入 DTMC 的转移概率矩阵可以由式 (4.14) 计算得到:

$$P = \begin{matrix} & \begin{matrix} RE \\ R \\ N \\ U \end{matrix} \\ \begin{matrix} RE \\ R \\ N \\ U \end{matrix} & \begin{bmatrix} 0 & p'_R & p'_N & 1 - p'_R - p'_N \\ 1 - p_{RU} & 0 & 0 & p_{RU} \\ 1 - p_{NU} & 0 & 0 & p_{NU} \\ 1 & 0 & 0 & 0 \end{bmatrix} \end{matrix} \quad (4.14)$$

式中, $p'_R = \frac{p_R}{p_R + p_N + p_U}$, $p'_N = \frac{p_N}{p_R + p_N + p_U}$ 。

由式 (4.7) 可以得到该简化模型嵌入 DTMC 各状态的稳态概率, 即

$$[\nu_{RE}, \nu_R, \nu_N, \nu_U] = \frac{1}{2 + p'_R p_{RU} + p'_N p_{NU}} [1, p'_R, p'_N, 1 - p'_R + p'_R p_{RU} - p'_N + p'_N p_{NU}] \quad (4.15)$$

已知 $\pi_i = \nu_i h_i / \sum_j \nu_j h_j$ 对任意 $i \in \{RE, R, N, U\}$ 成立, 因而可以得到简化的 SMP 模型的稳态概率表达, 即

$$[\pi_{RE}, \pi_R, \pi_N, \pi_U] = \alpha [h_{RE}, p'_R h_R, p'_N h_N, (1 - p'_R + p'_R p_{RU} - p'_N + p'_N p_{NU}) h_U] \quad (4.16)$$

式中,

$$\alpha = \frac{\nu_{RE}}{\sum_j \nu_j h_j} = \frac{1}{(2 + p'_R p_{RU} + p'_N p_{NU}) \sum_j \nu_j h_j} \quad (4.17)$$

1. 可靠性平均失效时间

为了探讨可信赖性属性的特性, 分别定义每个属性的平均失效时间。这里, 定义可靠性平均失效时间 (Mean Time To Reliability Failure, MTTRF) 为系统从可运行的正常状态开始, 直至进入可靠性不满足的状态所经过的平均时间。根据可靠性的定义, 定义可靠性失效状态构成 SMP 模型中的吸收态集合, 即 $X_a = \{R, N, U\}$; 可靠性满足的状态构成瞬态集合, 表示为 $X_t = \{A, IA, ID\} = \{RE\}$ 。这样, 在存在吸收态的 SMP 模型中, 转移概率矩阵可以通用地表示为

$$P = \begin{bmatrix} Q & C \\ 0 & I \end{bmatrix} \quad (4.18)$$

式中, 矩阵 Q 为瞬态之间的转移概率矩阵, 即

$$Q = \begin{bmatrix} 0 & p_{IA} & p_{ID} \\ p_A & 0 & p_{ID} \\ p_{IDA} & p_{IDIA} & 0 \end{bmatrix} \quad (4.19)$$

假设初始状态为瞬态中的可访问状态 (A), 形式化地表示为指示向量 $q = [1, 0, 0]$ 。则瞬态集合中每个状态的平均访问次数可由下式求解得到:

$$V_i = q_i + \sum_{j \in X_t} V_j Q_{ji} \quad (4.20)$$

结合式 (4.19) 和式 (4.20), 可以得到每个瞬态平均访问次数的解析表达, 即

$$V_A = \frac{1 - p_{ID}p_{IDIA}}{1 - p_Ap_{IA} - p_{ID}p_{IDA} - p_{IA}p_{ID}p_{IDA} - p_{ID}p_{IDIA} - p_Ap_{ID}p_{IDIA}} \quad (4.21)$$

$$V_{IA} = \frac{p_{IA} + p_{ID}p_{IDIA}}{1 - p_Ap_{IA} - p_{ID}p_{IDA} - p_{IA}p_{ID}p_{IDA} - p_{ID}p_{IDIA} - p_Ap_{ID}p_{IDIA}} \quad (4.22)$$

$$V_{ID} = \frac{p_{ID} + p_{IA}p_{ID}}{1 - p_Ap_{IA} - p_{ID}p_{IDA} - p_{IA}p_{ID}p_{IDA} - p_{ID}p_{IDIA} - p_Ap_{ID}p_{IDIA}} \quad (4.23)$$

从而, 由式 (4.24) 可以计算得到 MTTRF 的解析解:

$$MTTRF = \sum_{i \in X_t} V_i h_i = V_A h_A + V_{IA} h_{IA} + V_{ID} h_{ID} \quad (4.24)$$

另一方面, 若考虑简化的 SMP 模型, 则原模型中的三个瞬态 $\{A, IA, ID\}$ 聚合为一个宏状态 (RE), 从而瞬态转移概率矩阵 $Q = [1]$, 初始状态为 RE, 因此 $MTTRF = h_{RE}$ 。这样, 考虑到模型的等价性, 式 (4.25) 不仅给出了两种 SMP 模型中 MTTRF 的计算方法, 也揭示了简化的 SMP 模型中 h_{RE} 的计算方法。

$$MTTRF = V_A h_A + V_{IA} h_{IA} + V_{ID} h_{ID} = h_{RE} \quad (4.25)$$

前文中已经提到, 除 MTTRF 外, 可靠性可以由式 (4.13) 中的 $R(\tau)$ 作为评价标准。由于模型可以应对任意逗留时间分布, 因而模型分析中很难给出 $R(\tau)$ 的统一的表达。但是, 若已知可运行状态 (RE) 的逗留时间分布, 则可以得到 $R(\tau)$ 的解析式。例如, 假设 RE 状态的平均逗留时间服从参数为 $\lambda = 1/h_{RE}$ 的指数分布, 则发生失效时间的概率密度函数为 $f(x) = \lambda e^{-\lambda x}$, 累积概率分布函数为 $F(x) = 1 - e^{-\lambda x}$ 。那么, 系统可靠性可以由如下表达式计算得出:

$$R(\tau) = 1 - F(\tau) = e^{-\lambda \tau} = e^{-\tau/h_{RE}} \quad (4.26)$$

2. 可用性平均失效时间

可用性的平均失效时间 (Mean Time To Availability Failure, MTAF) 的计算方法比较直观、简单。由可用性定义可知, 瞬态仅包括可访问状态, 即 $X_t = \{A\}$, 转移概率矩阵为 $Q = [1]$, 因而瞬态的平均访问次数 $V_A = 1$ 。综上所述, 可用性的平均失效时间等于可访问状态的平均逗留时间, 即

$$MTAF = V_A h_A = h_A \quad (4.27)$$

3. 可维护性平均失效时间

若系统的可维护性失效，则代表系统发生了不可修复故障（N）或灾难性故障（U），无法通过自身的修复能力恢复至正常状态。所以，对于可维护性的平均失效时间（Mean Time To Maintainability Failure, MTMTF）分析，吸收态集合应为 $X_a = \{N, U\}$ ，对应的瞬态集合是 $X_t = \{RE, R\}$ 。瞬态间的转移矩阵为

$$Q = \begin{bmatrix} 0 & p'_R \\ 1 - p_{RU} & 0 \end{bmatrix} \quad (4.28)$$

规定初始状态为可运行状态（RE），即 $q = [1, 0]$ ，则各个瞬态的平均访问次数为

$$V_{RE} = \frac{1}{1 - p'_R + p'_R p_{RU}} \quad (4.29)$$

$$V_R = \frac{p'_R}{1 - p'_R + p'_R p_{RU}} \quad (4.30)$$

综上所述，MTMTF 可以通过下式得到：

$$MTMTF = \frac{1}{1 - p'_R + p'_R p_{RU}} \cdot h_{RE} + \frac{p'_R}{1 - p'_R + p'_R p_{RU}} \cdot h_R \quad (4.31)$$

4. 平均故障修复时间

在讨论系统的可维护性时，除稳态概率和 MTMTF 两种评价指标外，平均故障修复时间（Mean Time To Repair, MTTR）也是一个常用的参考指标，它代表了系统发生故障后需要的修复时间的期望值。采用上述类似的模型分析方法，从不同的建模视角出发，即可将本书中提出的 SMP 模型用于 MTTR 属性的评价。

这里，将系统的正常状态定义为吸收态，即 $X_a = \{RE\}$ ；将故障状态定义为瞬态，即 $X_t = \{R, N, U\}$ 。瞬态间的转移概率矩阵 Q 写为

$$Q = \begin{bmatrix} 0 & 0 & p_{RU} \\ 0 & 0 & p_{NU} \\ 0 & 0 & 0 \end{bmatrix} \quad (4.32)$$

MTTR 分析时，与前述的方法不同，需要定义故障状态作为初始状态。初始状态的指示向量为 $q = [\pi'_R, \pi'_N, \pi'_U]$ ，初始状态概率是一个条件概率，由 SMP 分析中得到的稳态概率得到，如：

$$\pi'_R = \frac{\pi_R}{\pi_R + \pi_N + \pi_U} \quad (4.33)$$

$$\pi'_N = \frac{\pi_N}{\pi_R + \pi_N + \pi_U} \quad (4.34)$$

$$\pi'_U = \frac{\pi_U}{\pi_R + \pi_N + \pi_U} \quad (4.35)$$

由式（4.20）可以得到 3 个故障状态的平均访问次数，即

$$V_R = \pi'_R \quad (4.36)$$

$$V_N = \pi'_N \quad (4.37)$$

$$V_U = p_{RU} \pi'_R + p_{NU} \pi'_N + \pi'_U \quad (4.38)$$

进而，用同样的方法，可以依据瞬态平均访问次数计算得到 MTTR 的解析式，即

$$\text{MTTR} = \frac{\pi_R}{\pi_R + \pi_N + \pi_U} \cdot h_R + \frac{\pi_N}{\pi_R + \pi_N + \pi_U} \cdot h_N + \left(\frac{p_{RU}}{\pi_R + \pi_N + \pi_U} + p_{NU} \frac{\pi_N}{\pi_R + \pi_N + \pi_U} + \frac{\pi_U}{\pi_R + \pi_N + \pi_U} \right) \cdot h_U \quad (4.39)$$

5. 保险性平均失效时间

进行保险性平均失效时间 (Mean Time To Safety Failure, MTTSF) 分析时, 吸收态集合仅包含灾难性故障一个状态, 即 $X_a = \{U\}$; 瞬态包括 $X_t = \{RE, R, N\}$; 转移矩阵 Q 为

$$Q = \begin{bmatrix} 0 & p'_R & p'_N \\ 1 - p_{RU} & 0 & 0 \\ 1 - p_{NU} & 0 & 0 \end{bmatrix} \quad (4.40)$$

设定初始状态为 RE, 则由式 (4.20) 得到瞬态平均访问次数为

$$V_{RE} = \frac{1}{1 - p'_R + p'_R p_{RU} - p'_N + p'_N p_{NU}} \quad (4.41)$$

$$V_R = \frac{p'_R}{1 - p'_R + p'_R p_{RU} - p'_N + p'_N p_{NU}} \quad (4.42)$$

$$V_N = \frac{p'_N}{1 - p'_R + p'_R p_{RU} - p'_N + p'_N p_{NU}} \quad (4.43)$$

进而可以得到服务计算系统的 MTTSF, 即

$$\text{MTTSF} = \frac{1}{1 - p'_R + p'_R p_{RU} - p'_N + p'_N p_{NU}} \cdot h_{RE} + \frac{p'_R}{1 - p'_R + p'_R p_{RU} - p'_N + p'_N p_{NU}} \cdot h_R + \frac{p'_N}{1 - p'_R + p'_R p_{RU} - p'_N + p'_N p_{NU}} \cdot h_N \quad (4.44)$$

6. 完整性平均失效时间

服务计算系统的完整性平均失效时间 (Mean Time To Integrity Failure, MTIIF) 的计算方法是, 设瞬态集合为 $X_t = \{A, IA\}$, 吸收态集合为 $X_a = \{ID, R, N, U\}$, 初始状态为可访问状态, 瞬态转移概率矩阵为

$$Q = \begin{bmatrix} 0 & p_{IA} \\ p_A & 0 \end{bmatrix} \quad (4.45)$$

则平均访问次数为

$$V_A = \frac{1}{1 - p_A p_{IA}}, \quad V_{IA} = \frac{p_{IA}}{1 - p_A p_{IA}} \quad (4.46)$$

进而可以计算得到 MTIIF, 即

$$\text{MTIIF} = \frac{1}{1 - p_A p_{IA}} \cdot h_A + \frac{p_{IA}}{1 - p_A p_{IA}} \cdot h_{IA} \quad (4.47)$$

4.3 可信赖性属性之间的相互关系

在前述的可信赖性属性的量化分析基础上, 本节进一步对得到的结果进行探讨。下面将

通过 3 个定理的形式化描述和理论证明, 讨论可信赖性属性之间的相互关系, 包括比较关系和相互影响关系。

定理 4.1 揭示了不同可信赖性属性的平均失效时间之间的比较关系。

【定理 4.1】可信赖性属性的平均失效时间之间存在如下的比较关系:

$$\text{MTTSF} \geq \text{MTTMF} \geq \text{MTTRF} \geq \text{MTTIF} \geq \text{MTTAF} \quad (4.48)$$

证明: (1) $\text{MTTSF} \geq \text{MTTMF}$ 。

考虑概率的取值范围, 有 $0 \leq p_{\text{NU}} \leq 1$, $0 \leq p'_{\text{R}} \leq 1$ 和 $0 \leq p_{\text{RU}} \leq 1$ 。因此, 可以推导得出:

$$1 - p'_{\text{R}} + p'_{\text{R}}p_{\text{RU}} \geq 1 - p'_{\text{R}} + p'_{\text{R}}p_{\text{RU}} - p'_{\text{N}}(1 - p_{\text{NU}}) \geq 0$$

所以

$$\frac{1}{1 - p'_{\text{R}} + p'_{\text{R}}p_{\text{RU}} - p'_{\text{N}}(1 - p_{\text{NU}})} \geq \frac{1}{1 - p'_{\text{R}} + p'_{\text{R}}p_{\text{RU}}} \geq 0$$

此外, 有:

$$\frac{p'_{\text{N}}}{1 - p'_{\text{R}} + p'_{\text{R}}p_{\text{RU}} - p'_{\text{N}}(1 - p_{\text{NU}})} \cdot h_{\text{N}} \geq 0$$

因此可以得到:

$$\begin{aligned} \text{MTTSF} &= \frac{1}{1 - p'_{\text{R}} + p'_{\text{R}}p_{\text{RU}} - p'_{\text{N}}(1 - p_{\text{NU}})} \cdot h_{\text{RE}} + \\ &\quad \frac{p'_{\text{R}}}{1 - p'_{\text{R}} + p'_{\text{R}}p_{\text{RU}} - p'_{\text{N}}(1 - p_{\text{NU}})} \cdot h_{\text{R}} + \\ &\quad \frac{p'_{\text{N}}}{1 - p'_{\text{R}} + p'_{\text{R}}p_{\text{RU}} - p'_{\text{N}}(1 - p_{\text{NU}})} \cdot h_{\text{N}} \\ &\geq \frac{1}{1 - p'_{\text{R}} + p'_{\text{R}}p_{\text{RU}}} \cdot h_{\text{RE}} + \frac{p'_{\text{R}}}{1 - p'_{\text{R}} + p'_{\text{R}}p_{\text{RU}}} \cdot h_{\text{R}} = \text{MTTMF} \end{aligned}$$

(2) $\text{MTTMF} \geq \text{MTTRF}$ 。

由前述证明过程可知:

$$\frac{1}{1 - p'_{\text{R}}(1 - p_{\text{RU}})} \geq 1, \quad \frac{p'_{\text{R}}}{1 - p'_{\text{R}} + p'_{\text{R}}p_{\text{RU}}} \cdot h_{\text{R}} \geq 0$$

因此有:

$$\text{MTTMF} = \frac{1}{1 - p'_{\text{R}}(1 - p_{\text{RU}})} \cdot h_{\text{RE}} + \frac{p'_{\text{R}}}{1 - p'_{\text{R}} + p'_{\text{R}}p_{\text{RU}}} \cdot h_{\text{R}} \geq h_{\text{RE}} = \text{MTTRF}$$

(3) $\text{MTTRF} \geq \text{MTTIF}$ 。

显然, $0 \leq p_{\text{ID}}p_{\text{IDIA}} < 1$, 进而得到:

$$\begin{aligned} &\frac{1 - p_{\text{ID}}p_{\text{IDIA}}}{1 - p_{\text{A}}p_{\text{IA}} - p_{\text{ID}}p_{\text{IDA}} - p_{\text{IA}}p_{\text{ID}}p_{\text{IDA}} - p_{\text{ID}}p_{\text{IDIA}} - p_{\text{A}}p_{\text{ID}}p_{\text{IDIA}}} \\ &= \frac{1}{1 - \frac{p_{\text{A}}p_{\text{IA}}}{1 - p_{\text{ID}}p_{\text{IDIA}}} - \frac{p_{\text{ID}}p_{\text{IDA}} + p_{\text{IA}}p_{\text{ID}}p_{\text{IDA}} + p_{\text{A}}p_{\text{ID}}p_{\text{IDIA}}}{1 - p_{\text{ID}}p_{\text{IDIA}}}} \\ &\geq \frac{1}{1 - p_{\text{A}}p_{\text{IA}}} \end{aligned}$$

又由 $p_{\text{IA}} + p_{\text{ID}}p_{\text{IDIA}} \geq p_{\text{IA}}$ 和 $1 - p_{\text{A}}p_{\text{IA}} - p_{\text{ID}}p_{\text{IDA}} - p_{\text{IA}}p_{\text{ID}}p_{\text{IDA}} - p_{\text{ID}}p_{\text{IDIA}} - p_{\text{A}}p_{\text{ID}}p_{\text{IDIA}} \leq 1 - p_{\text{IA}}p_{\text{IDIA}}$,

可知:

$$\frac{p_{IA} + p_{ID}p_{IDA}}{1 - p_A p_{IA} - p_{ID}p_{IDA} - p_{IA}p_{ID}p_{IDA} - p_{ID}p_{IDA} - p_A p_{ID}p_{IDA}} \geq \frac{p_{IA}}{1 - p_A p_{IA}}$$

综上所述, 结合式 (4.24) 和式 (4.47) 中 MTTRF 和 MTTIF 的解析式, 可得如下结论:

$$\text{MTTRF} \geq \text{MTTIF}$$

(4) $\text{MTTIF} \geq \text{MTTAF}$ 。

考虑到已有 $\frac{1}{1 - p_A p_{IA}} \geq 1$ 和 $\frac{p_{IA}}{1 - p_A p_{IA}} \cdot h_{IA} \geq 0$, 可以证明:

$$\text{MTTIF} = \frac{1}{1 - p_A p_{IA}} \cdot h_A + \frac{p_{IA}}{1 - p_A p_{IA}} \cdot h_{IA} \geq h_A = \text{MTTAF}$$

综上所述, 可以得到式 (4.48) 所示结论, 定理得证。

定理 4.1 揭示了可信赖性属性 MTTF 之间的比较关系, 它直观显示出, 在一般服务计算系统中, 可用性的平均失效时间最短; 由于可用性对系统状态的要求较为严格, 因此该属性比其他属性更易被破坏; 而保险性一般可以保证的时间在 5 个属性中相对最长, 一般而言, 系统发生灾难性后果的概率比较低, 保险性更易得到保证。

从稳态概率分析的角度来看, 可信赖性属性之间也存在一些相互影响的关系。定理 4.2 和定理 4.3 将给出这些关系的量化表达, 揭示属性之间的内在规律。

【定理 4.2】 从稳态分析意义上, 可维护性的提升可以导致保险性的改善。进一步, 可维护性对应的模型状态是保险性的子集。

证明: 由可维护性的定义可知, 其量化表达可以写作:

$$M = \Pr(R | \text{Failed})$$

对应地, 保险性的表达式可以作如下的变形:

$$S = \Pr(\text{Safe} | \text{Failed}) = \Pr(R | \text{Failed}) + \Pr(N | \text{Failed}) = M + \Pr(N | \text{Failed})$$

因此, 可维护性 M 的提高将带来保险性 S 的增大。

另外, 从图 4.1 中可以发现, 保险性保证的状态是一个聚集状态, 包含了可修复故障 (R) 和不可修复故障 (N) 两个子状态。而可维护性对应的状态是其中的可修复故障状态。因此可以得到结论: 可维护性对应的模型状态是保险性状态的子集。

【定理 4.3】 在其他因素保持不变的情况下, 可用性的改善可以促进完整性的提高。

证明: 由完整性的定义式 (4.10) 推导得到:

$$I = \frac{\pi_A + \pi_{IA}}{\pi_A + \pi_{IA} + \pi_D} = 1 \left/ \left(1 + \frac{\pi_{ID}}{\pi_A + \pi_{IA}} \right) \right.$$

已知可用性的表达为 $A = \pi_A$, 从而定理 4.3 得证。

另一方面, 这些定理可以由数值实验得以验证。图 4.3 所示为可信赖性属性随着参数 p_R 的变化情况。从分析过程中可以看出, p_R 主要影响系统的可维护性。从图中可以明显看出, 随着 p_R 的增加, 系统的可维护性显著提高, 并且带动了保险性的提升, 从而验证了定理 4.2。在图 4.4 所示的实验中, 提高参数 h_A , 即提高系统的可用性, 可以在一定程度上增大系统的完整性, 验证了定理 4.3 的结论。此外, 实验结果给出了随着 h_A 的增大各属性 MTTF 的变化情况, 如图 4.5 所示。该实验不仅验证了定理 4.1 中的 MTTF 比较关系, 也从

侧面上验证了模型分析的正确性和合理性。

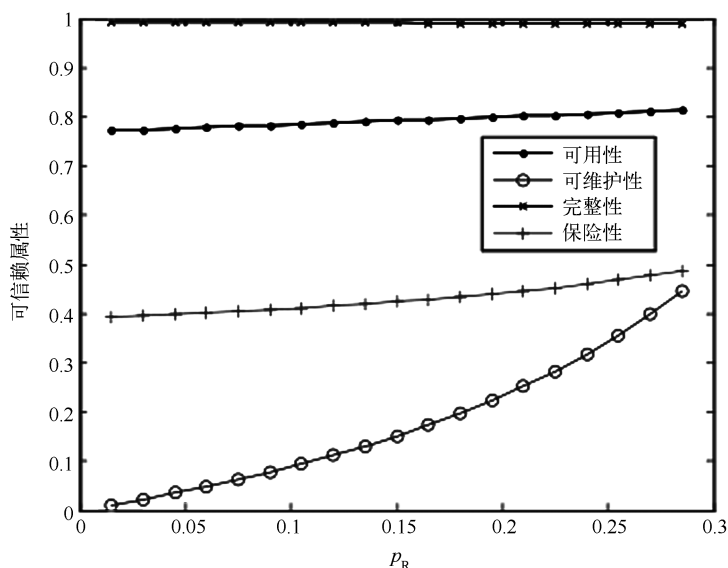


图 4.3 可信赖性属性随参数 p_R 的变化情况

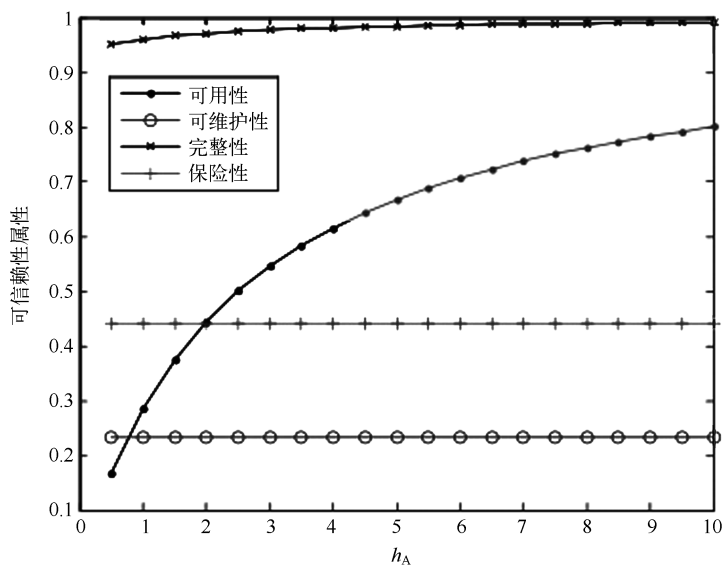
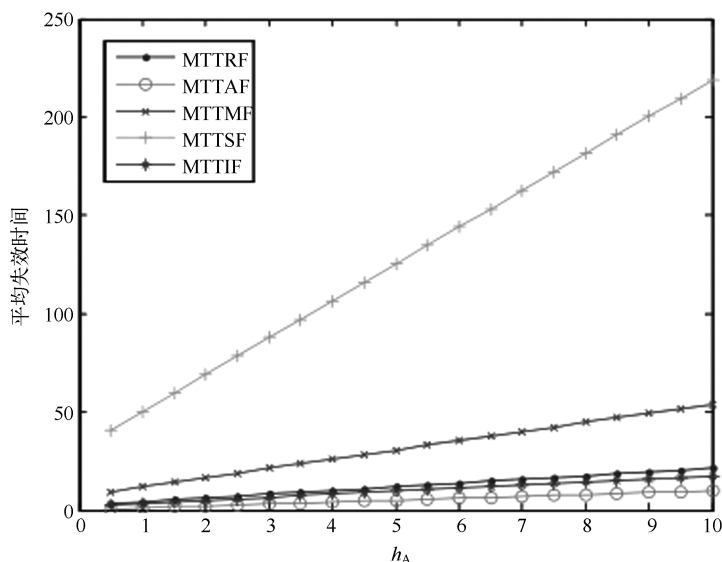


图 4.4 可信赖性属性随参数 h_A 的变化情况

这些定理揭示了可信赖性属性之间的相互关系，对系统的设计和优化具有一定的参考价值。例如，通过改进系统设计和优化系统机制，将系统故障尽量控制在可修复故障的范围内，而避免发生不可修复或灾难性故障，可以同时提高服务系统的可维护性和保险性。又如，通过在系统设计中引入改善可用性的相关技术和机制，可以同时带来完整性的提升。这些结论可以为服务计算系统的设计和优化提供理论指导和支撑。

图 4.5 可信赖性属性的 MTTF 随参数 h_A 的变化情况

4.4 服务选择和服务组合的可信赖性建模与分析

考虑到服务计算中动态性特点，在可信赖性分析中，需要对其中的动态服务组合和服务选择进行分析评价。开展服务选择和服务组合的可信赖性建模与分析，可以揭示服务计算动态性对服务可信赖性带来的影响。

服务可以根据任务需求分为简单服务和复杂服务。简单服务一般具有某一特定的功能，并且可以相互协同，构成复杂服务。在当今的业务需求中，业务流程往往需要一系列的服务同时或按照一定顺序协同地完成多样化的任务，并最终进行服务整合。服务组合是指从可用的服务集合中选取满足用户需求、保障服务质量的一系列服务，并按照业务流程组合成为一个复杂服务工作流的过程^[2]。

4.4.1 基本框图模式的可信赖性分析

本节将借鉴可靠性框图（Reliability Block Diagram, RBD）的思路，建立服务组合的可信赖性分析模型。在 RBD 理论体系中，有两种最基本的框图模式，即串行（Series）模式和并行（Parallel）模式。前者代表满足功能需求的服务串行连接，需要全部执行完成才能实现所需的功能；而后者代表服务并行执行，相互可以作为备份，只要其中之一正常工作，即可满足用户的功能需求。

对于可信赖性分析，串行模式的服务必须同时处于正常状态才能完成整个任务。假设对于每个简单服务 i 需要分析的可信赖性属性（稳态分析）为 $D_i(t)$ （它可以代表该服务的可靠性、可用性、可维护性、保险性或完整性中的任意属性），则串行组合的服务的可信赖性 $D(t)$ 可通过下式得到：

$$D(t) = \prod_i D_i(t) \leq \min_i \{D_i(t)\} \quad (4.49)$$

若简单服务 i 的某一可信赖性 MTTF 定义为 MTTF_i ，则组合的复杂服务的 MTTF 可以通过下式得到：

$$\text{MTTF} = 1 / \sum_i \frac{1}{\text{MTTF}_i} \leq \min_i \{ \text{MTTF}_i \} \quad (4.50)$$

考虑到 $D_i(t)$ 是稳态概率分析，有 $0 \leq D_i(t) \leq 1$ ，则串行组合服务的可信赖性 $D(t)$ 和 MTTF 将比构成其工作流的任意简单服务的都小。

另一方面，对于并行服务模式，其可信赖性分析如下：

$$D(t) = 1 - \prod_i (1 - D_i(t)) \geq \max_i \{ D_i(t) \} \quad (4.51)$$

$$\begin{aligned} \text{MTTF} &= \sum_i \text{MTTF}_i - \sum_{i \neq j} \frac{1}{\frac{1}{\text{MTTF}_i} + \frac{1}{\text{MTTF}_j}} + \sum_{i \neq j \neq k} \frac{1}{\frac{1}{\text{MTTF}_i} + \frac{1}{\text{MTTF}_j} + \frac{1}{\text{MTTF}_k}} \dots \\ &\geq \max_i \{ \text{MTTF}_i \} \end{aligned} \quad (4.52)$$

可以证明，并行模式的可信赖性比其任意简单服务的都要大。

4.4.2 基本服务模式的可信赖性分析

服务的动态组合可以抽象为一些典型的工作流服务模式^[86]。本节从 5 个基本的服务模式开展研究，这些模式可以描述服务组合流程控制中的常用基本元素。更为复杂的服务模式可以由这些基本模式组合构造而成，并通过类似的分析方法加以评价。

1. 串行序列模式

考虑两个服务的服务组合，一个服务需要在另一个服务执行完成后才能被激活，这样的服务组合称为串行序列模式（Sequence Pattern），如图 4.6（a）所示。

在这类服务模式中，必须要求两个服务皆正常工作，才能保证该组合服务的有效性。设这两个服务的可信赖性属性分别为 $D_A(t)$ 和 $D_B(t)$ ，该属性的平均失效时间分别是 MTTF_A 和 MTTF_B ，则组合服务的可信赖性属性及其 MTTF 为

$$D(t) = D_A(t) \cdot D_B(t) \quad (4.53)$$

$$\text{MTTF} = \frac{\text{MTTF}_A \cdot \text{MTTF}_B}{\text{MTTF}_A + \text{MTTF}_B} \quad (4.54)$$

2. 并行分支模式

在并行分支服务模式（Parallel Split Pattern）中，某一线程在分支点分为两个可以同步执行的分支线程，如图 4.6（b）所示。虽然两个分支线程的执行过程可以并行化，但最终结果需要两个线程皆完成服务过程才能得到。因此，这两个线程在框图模型中的关系仍是串行的。因此，该服务组合模式的可信赖性稳态值和 MTTF 计算方法为：

$$D(t) = D_A(t) \cdot D_B(t) \cdot D_C(t) \quad (4.55)$$

$$\text{MTTF} = 1 / \left(\frac{1}{\text{MTTF}_A} + \frac{1}{\text{MTTF}_B} + \frac{1}{\text{MTTF}_C} \right) \quad (4.56)$$

3. 并行同步模式

并行同步模式（Synchronization Pattern）是指多个服务子过程需要将输出结果进行同步聚合后，再输入给某一个后续服务，如图 4.6（c）所示。所以，并行同步模式的框图模型与并行分支模式完全相同，该组合模式的可信赖性属性也可以由式（4.55）来评价，MTTF

也可量化表达为式 (4.56)。

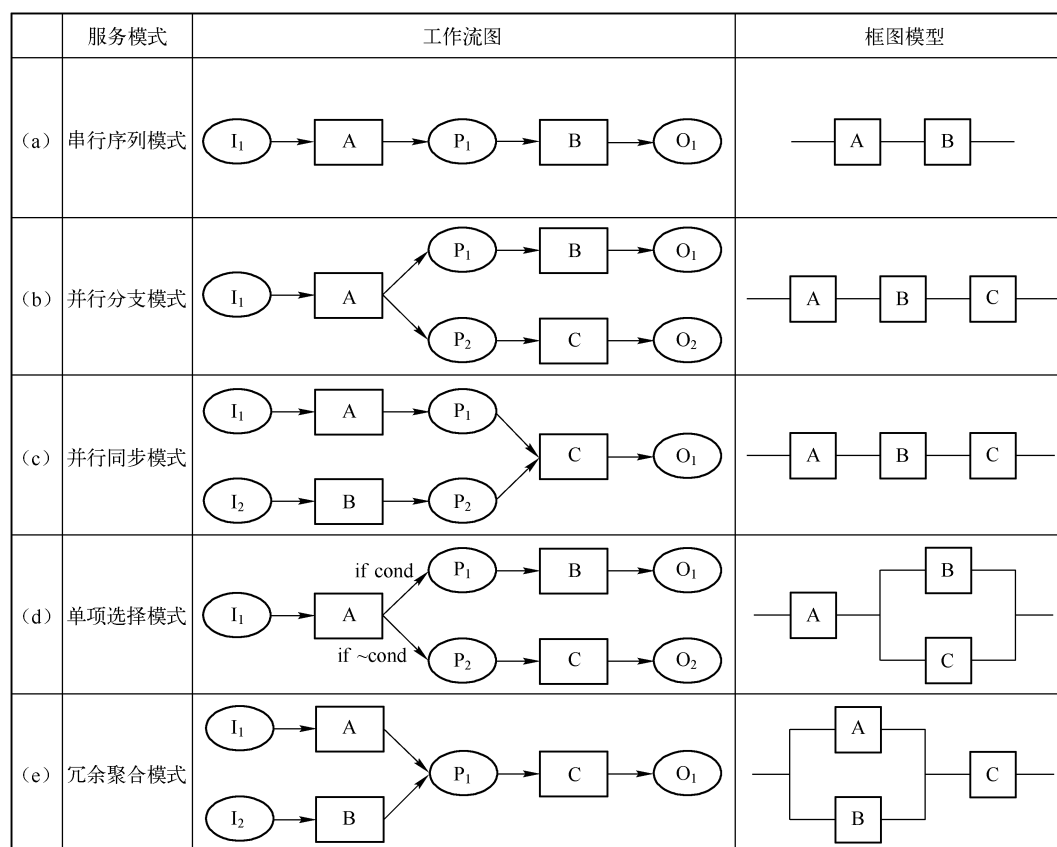


图 4.6 基本服务模式及其分析框图

4. 单项选择模式

单项选择模式 (Exclusive Choice Pattern) 又被称为服务选择 (Service Selection), 旨在根据一定的控制条件, 选择多个服务中的一个完成服务过程。单项选择模式的示意图如图 4.6 (d) 所示。由于待选服务集合中只要有一个服务执行完成即可, 因而从框图分析的角度来看, 集合中的服务之间是并行的关系。单项选择模式的可靠性和 MTTF 计算公式为

$$D(t) = D_A(t) \cdot [1 - (1 - D_B(t))(1 - D_C(t))] \quad (4.57)$$

$$MTTF = 1 / \left(\frac{1}{MTTF_A} + \frac{MTTF_B + MTTF_C}{MTTF_B^2 + MTTF_B \cdot MTTF_C + MTTF_C^2} \right) \quad (4.58)$$

5. 冗余聚合模式

冗余聚合模式 (Simple Merge) 代表多个冗余备份服务分支在某一点聚合, 输出结果供后续服务使用, 聚合中不需要同步过程。该模式的示意图和分析框图如图 4.6 (e) 所示。冗余聚合模式的可靠性和 MTTF 计算公式为

$$D(t) = [1 - (1 - D_A(t))(1 - D_B(t))] \cdot D_C(t) \quad (4.59)$$

$$MTTF = 1 / \left(\frac{MTTF_A + MTTF_B}{MTTF_A^2 + MTTF_A \cdot MTTF_B + MTTF_B^2} + \frac{1}{MTTF_C} \right) \quad (4.60)$$

4.5 案例分析

为了揭示本章提出的建模分析理论在实际系统中的适用范围和应用方法,本节将在实际系统中应用前述模型,并给出案例分析。进一步,本节将在实际数据集上进行参数敏感度分析,确定系统瓶颈,以指导系统的改进。

4.5.1 系统简介

为了在真实系统中应用本章提出的模型,并对模型参数和评价结果进行讨论,本节选用了美国洛斯阿拉莫斯国家实验室(Los Alamos National Laboratory, LANL)提供的大规模系统故障日志^[85]。该日志描述的系统是LANL管理维护的23个高性能计算系统(包含4750个服务器、24101个处理器)。本节选取其中的19个提供计算服务、网格服务或Web服务的系统,将其作为分析对象。它们承载了各种应用服务,并且具有大规模架构、动态服务共享与组合等服务计算的显著特性。该系统日志中记录了从1996年至2005年这些系统中发生的各种错误、故障和修复的情况。

LANL将这些系统发生的故障在日志中分为6个类别,即硬件故障(Hardware Failure)、软件故障(Software Failure)、设备故障(Facilities Failure)、网络故障(Network Failure)、人为失误(Human Error)和未知错误(Undetermined)。其中,软件故障和人为失误是比较易于处理的故障,大多数可以通过虚拟机自动迁移和重启等手段得以自动修复,因此可将其归类为SMP模型中的可修复故障;设备故障代表了周边设备出现了失效,包括电源故障、外界停电等事件,这类故障无法通过系统自身进行修复,但并非是系统内部的严重故障,因此被视为不可修复故障;网络故障主要影响系统与外界的交互渠道,这类故障会使系统转至不可访问的状态;硬件故障代表系统内部发生了关键组件的失效,本节将灾难性故障定义为硬件故障,代表此类故障将直接中断服务过程,造成服务失效,并且需要巨大的人力、物力开销才可修复。

4.5.2 可信赖性分析

通过统计方法对LANL系统故障日志进行分析,可以发现,故障发生的平均间隔时间服从韦伯分布(Weibull Distribution),而故障修复时间近似服从对数正态分布(Lognormal Distribution)^[59]。因此,一些传统的需要假设状态平均逗留时间为指数分布的模型方法^[38,79,87]可能无法得到准确的分析结果,因而不适用。而本章提出的模型由于不需要任何状态逗留时间的假设或先验知识,因而能很好地适用于该数据分析。

由于该系统日志中尚无系统和数据完整性的记录,因此无法对该系统的完整性进行评价。前文描述中已指出,为了应对此类数据缺失问题,可以采用简化的SMP模型进行可信赖性评价。这里需要指出的是,在某些服务计算系统中,尤其是数据存储敏感型系统,完整性是一个重要的需求。而在系统日志中,必然记录相关的事件日志,因此在这类系统中就可以应用图4.1所示的SMP模型对包括完整性在内的可信赖性属性进行评价。

首先可以根据系统日志中各事件发生的时间获得系统状态转移序列,并利用统计方法得到SMP模型中的状态转移概率和平均逗留时间等模型参数;然后对SMP模型进行求解,得

到各个状态的稳态概率，进而根据可信赖性属性的形式化定义，得到可信赖性属性的评价结果。本节对选取的19个系统进行建模和分析，得到了它们的可维护性和保险性评价结果，如图4.7所示。

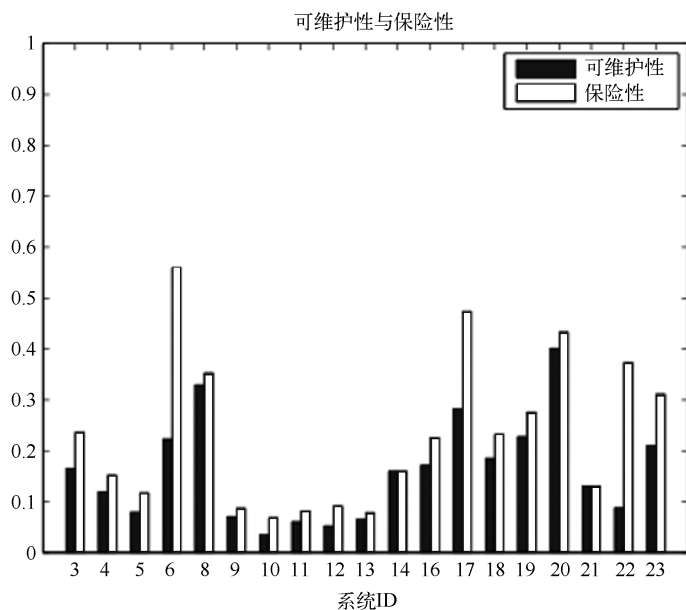


图 4.7 19 个系统的可维护性和保险性评价结果

根据统计得到的模型参数和本章所提出的分析方法，可以得到系统可靠性、可维护性、保险性的平均失效时间 (MTTF)，以及系统的平均修复时间 (MTTR)，如图4.8和图4.9所示。该分析结果验证了定理4.1的正确性。

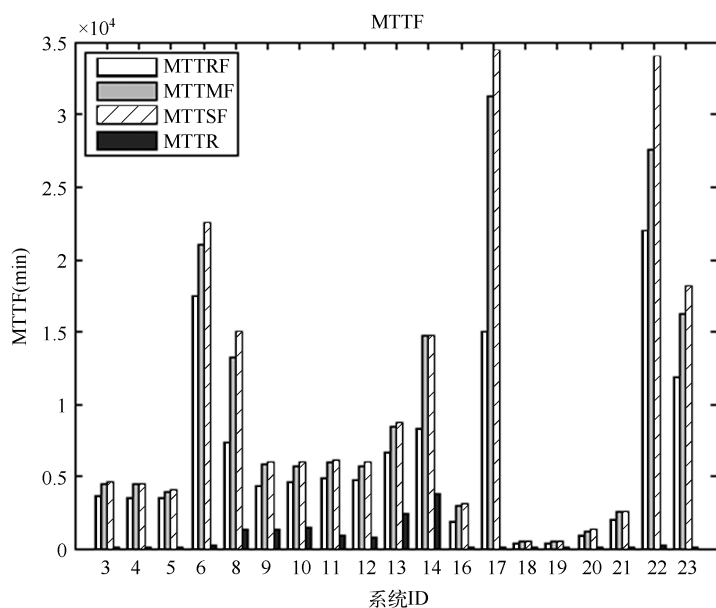


图 4.8 19 个系统的 MTTF 评价结果

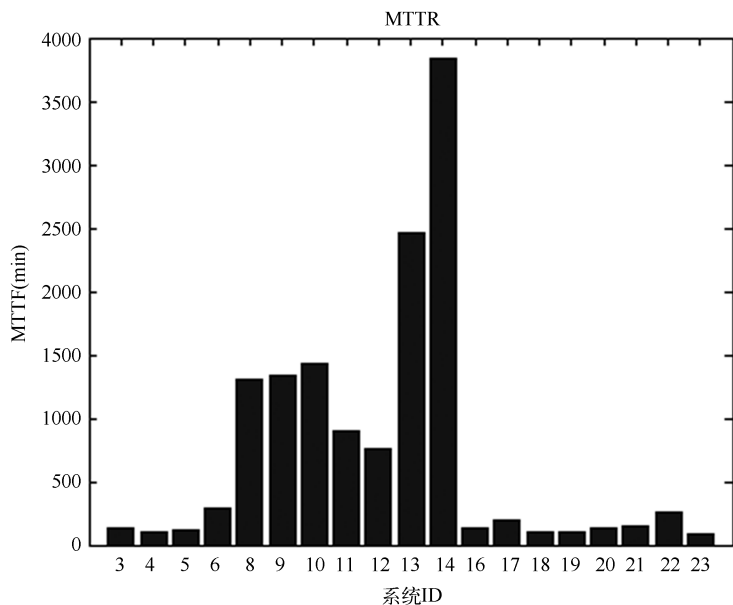


图 4.9 19 个系统的 MTTR 评价结果

从实验结果中可以看出，不同系统的可信赖性区别明显。造成这一现象的主要原因是，不同的系统类型和不同的负载情况造成了模型参数的巨大差异^[59]。由此可见，系统参数对可信赖性评价指标的影响将对系统设计和改进具有重要的参考价值。

4.5.3 敏感度分析

敏感度分析（Sensitivity Analysis）是探讨模型参数对评价结果的影响，用于发现系统或模型中对错误最为敏感的部件，找出系统瓶颈^[88]。本节在 SMP 模型基础上进行参数敏感度分析，确定模型参数对可信赖性属性及其 MTTF 的影响程度，用于发现影响度（即敏感度）最大的参数，以指导改进路线设计。

假设 Y 是进行敏感度分析的评价指标，它可以是可信性属性，也可以是 MTTF 或 MTTR。 Y 与多个系统参数 λ_i 有关，则 Y 对于参数 λ_i 的敏感度 $S_{\lambda_i}(Y)$ 的定义为

$$S_{\lambda_i}(Y) = \left| \frac{\partial Y}{\partial \lambda_i} \right| \quad (4.61)$$

在工程实践中，为了简化分析过程，降低求解开销，往往借助于 λ_i 的无穷小量 $\Delta\lambda_i$ 进行敏感度分析。因而， $S_{\lambda_i}(Y)$ 可以由下式计算得到：

$$S_{\lambda_i}(Y) = \left| \frac{Y(\lambda_i + \Delta\lambda_i) - Y(\lambda_i)}{\Delta\lambda_i} \right| \quad (4.62)$$

本节在 LANL 系统日志的基础上进行可信赖性和 MTTF、MTTR 的参数敏感度分析。图 4.10 所示的是对状态平均逗留时间参数的敏感度分析结果，图 4.11 所示的是对状态转移概率参数的敏感度分析结果。图中纵轴值的大小反映了该参数对评价指标的影响程度，该值最大的参数即为系统在这一评价指标上的瓶颈。例如，图 4.10 中可见，系统在可运行状态的平均逗留时间 h_{RE} 是对 MTTRF、MTTME 和 MTTSF 影响最大的系统参数，意味着若想提高这些指标，可以引入一些可靠硬件保障技术，减少故障发生，延长系统在可运行状态的停留

时间。而从实验结果可以看出, h_{RE} 对 MTTR 的影响却微乎其微, 直观上可以理解, 延长组件寿命, 减少故障发生, 对于故障的修复过程不会造成显著影响。另外, 从敏感度分析结果可见, h_R 和 h_U 两个参数是优化可维护性和保险性的关键因素。从图 4.11 中可以看出, 提升 p'_R 对可维护性、MTTMF 和 MTTSF 的改善具有重要作用, 即减少不可修复和灾难性故障的发生概率是改善可维护性的有效方式。而减少 p'_N 是降低 MTTR 的有效手段, 从数据中可以发现, 对于不可修复故障, 其修复时间在绝大多数系统中都非常可观, 因而避免进入该状态可以有效提高 MTTR。

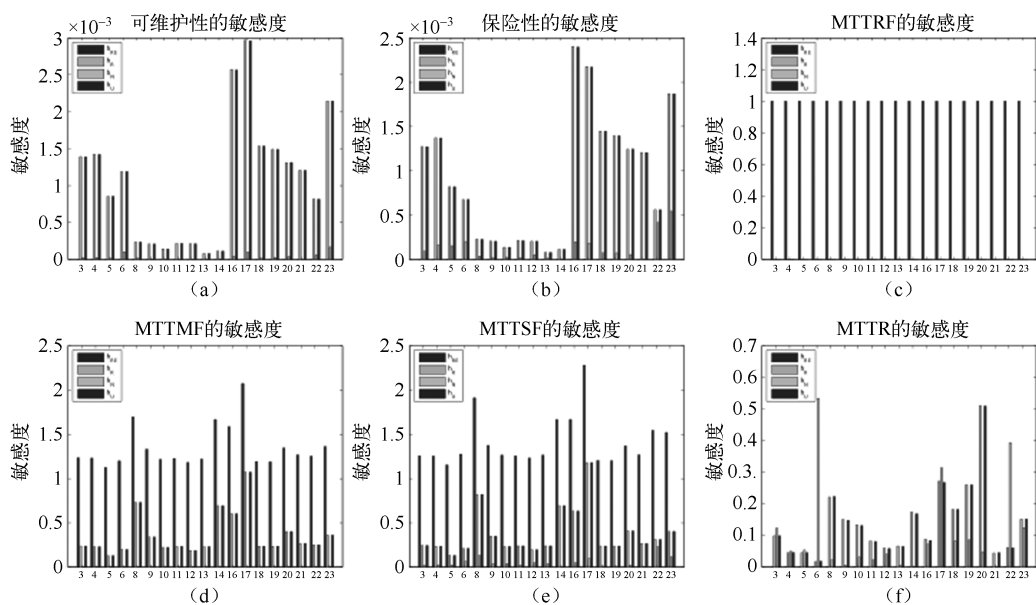


图 4.10 对状态平均逗留时间参数的敏感度分析结果

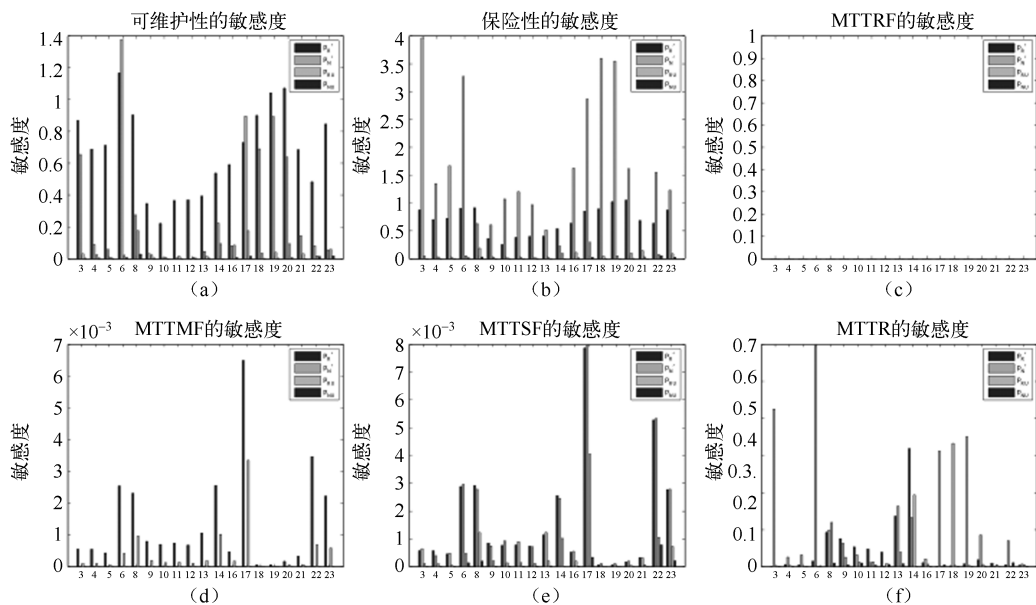


图 4.11 对状态转移概率参数的敏感度分析结果

注意，在模型分析中， $MTTRF = h_{RE}$ ，（参见式（4.25））。显然， $MTTRF$ 对 h_{RE} 的敏感度为 1，对其他任意参数的敏感度均为 0。这一结论也可以从图 4.10（c）和图 4.11（c）中得以验证。实验结果在一定程度上验证了本章所建立模型的合理性和正确性。

4.6 本章小结

本章从面向服务的角度出发，全面、系统地研究了服务计算系统中可信赖性的建模方法和分析技术。本章提出了服务计算系统的半马尔可夫状态转移模型，给出了可信赖性属性的形式化定义。本章给出了模型的分析求解方法及状态聚合方法，并基于此给出了可信赖属性的量化表达，包括稳态概率分析，以及各属性的平均失效时间分析。在此基础上，本章研究了可信赖性属性之间的相互关系，给出了数学表达。为了应对服务动态性，本章给出了典型的服务组合模型的可信赖性评价模型和分析方法。最后，在大规模的实际系统故障日志数据上开展案例分析，并进行了敏感度分析，以发现系统瓶颈，指导优化方案的制定与选择。

基于随机游走模型的服务质量排序评价

在前述讨论中，主要关注的是对单个服务计算系统或该系统所提供服务的服务质量的评价；而在实际的应用场景中，有时还需要对整个互联网或其中的一个子网范围内的服务同时进行评价。例如，当某些旅行代理商需要处理用户订购机票的任务时，需要调用各个航空公司的订票服务，将结果进行整合和处理后，供用户选择和使用，因此需要对各个公司的订票服务进行评价和比较，以呈现给用户更符合其需求的结果。目前，对于这样的场景，绝大多数的代理商都采用了遍历评价的方法，即每个代理商对于其需要的所有服务都进行评价，然后进行数据处理和排序。随着互联网上的服务数目逐渐增加，这样的评价方法越来越受到规模的制约，无论是进行模型评价和实际测量，都会遇到评价复杂度过高和评价成本过大的问题。因此，大规模服务评价是服务计算中亟待解决的问题。

在服务计算中，服务评价的主要目的是对不同的服务进行比较，其典型的应用场景为服务组合或服务推荐。当确定待评价的服务集合（即满足用户特定需求的一组服务集合）后，代理商将根据这些服务的多属性的 QoS 值给出排序好的结果清单^[89]。在这样的应用背景下，相比 QoS 各属性的绝对值，代理商更关注于服务排序的结果。一方面，该排序过程可以通过一些评价函数对多个 QoS 属性进行聚合，随后按照函数值进行排序；另一方面，也可直接通过服务之间的两两比较得到排序的结果。考虑到评价函数的标准在不同代理商之间可能存在不一致的情况，且评价时由于环境的差异可能导致 QoS 绝对值的偏差（如测量误差），因而在进行排序时，基于比较的方法往往可以给出更客观的符合内在规律的排序结果^[90]。

本章针对服务评价中的比较排序问题，研究大规模服务环境中的高效的服务评价技术。针对任意的单个服务评价方法（包括前述的基于模型分析的方法，或者实际测量方法和预测方法等），本章提出了一种协同分布式的服务评价框架和方法。该方法可以避免单个测试端（代理商）对所有服务进行评价所带来的巨大开销。在该评价框架中，每个测试端只需要测试部分服务集合，随后由一个集中控制器将结果进行聚合，即可得到全局的比较结果。本章将建立基于比较的服务排序随机模型，利用随机游走和马尔可夫链理论对模型进行分析求解，并在此基础上，提出将部分排序结果聚合成全局排序的算法，给出算法的理论分析和实验验证。本章提出的算法可以有效容忍不同测试端排序过程中存在的误差，为服务选择、服务推荐等过程提供准确、鲁棒的排序结果。通过部分排序进行聚合的评价框架，可以显著降低每个测试端的评价开销，为大规模服务评价提供技术保障。此外，该方法还可以应用于基于预测的服务评价^[91]中，可以进一步降低训练集的规模，提高预测的效率。

5.1 协同分布式服务评价框架

本章针对如何通过分布式部分服务的评价排序聚合得到全局排序的问题，研究排序聚合模型、方法和技术，旨在减少测试端的评价开销，提高评价的鲁棒性，为大规模服务评价提供有效的理论和技术支撑。考虑到排序的普适性，本章中所提出的排序评价方法可以建立在任意的服务评价方法之上，包括模型方法、测量方法和预测方法，具有广泛的适用范围。另外，在排序过程中将考虑到分布式排序过程中可能存在的误差，应用概率论、随机过程和数理统计等相关理论，给出最为符合真实情况的全局排序结果。

本章讨论的基于部分排序的协同分布式服务评价框架如图 5.1 所示。设需要评价（排序）的所有服务数目为 n ，将它们按顺序编号，表示为 $S = \{1, 2, \dots, n\}$ 。设整个评价框架中共有 m 个分布式的测试端，每个测试端仅对整个服务集合中的某个服务子集进行评价和排序。这里将第 j 个测试端排序得到的服务子集表示为 $S_j \subseteq S$ ， S_j 中的每个服务 $s \in S_j$ 在排序后得到排序值定义为 $r_j(s) \in \{1, 2, \dots, |S_j|\}$ ，从而 S_j 中的所有服务排序结果可以表示为集合 $R_j = \{\langle s, r_j(s) \rangle \mid s \in S_j\}$ 。在该框架中，每个测试端维护了部分服务的排序结果。中心化的集中控制器将收集所有测试端的排序结果，并将它们进行处理和聚合，从而得到全局排序。针对整个服务排序和聚合的过程，本章的余下各节将分别从模型、算法和分析等方面分别展开讨论。

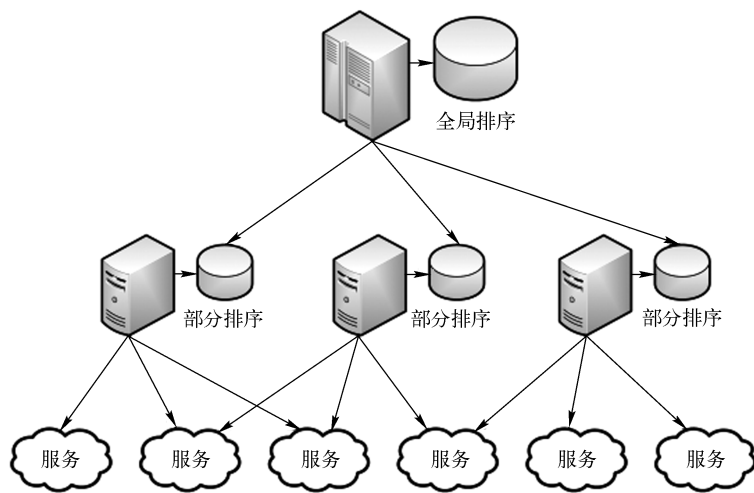


图 5.1 基于部分排序的协同分布式服务评价框架

5.2 服务评价排序模型

为了深入认识排序过程，本节将对比较排序进行建模研究。首先对服务之间的比较过程建立概率模型，随后将服务排序形式化为随机游走过程。进一步，该随机游走可以对应到离散时间马尔可夫链（DTMC）中，从而通过求解该 DTMC 稳态概率得到排序结果。

5.2.1 基于比较的概率模型

由于排序过程的本质即为服务之间的两两比较,因而首先对服务比较进行建模。这里采用概率模型描述服务之间的两两比较关系^[92,93]。在模型中,假设每个服务 $i \in S$ 本身具有一个权值 $w_i \in \mathbb{R}^+$,这个权值本质上由该服务的 QoS 来决定,是服务固有的一个特性。在此不必知道该权值是如何计算得到的,也无须规定该权值的评价标准。当对服务集合中的两个服务 (i,j) 进行比较时,认为该比较的结果由两个服务各自的权值决定。考虑到比较中存在的误差和不确定性,将比较的输出结果 X_{ij}^l 定义为一个随机变量,其中 $X_{ij}^l = 1$ 表示服务 j 在比较中胜出(即排名比 i 高),反之 $X_{ij}^l = 0$ 代表服务 i 胜出; l 代表 (i,j) 比较的次数。进一步,假设所有随机变量 X_{ij}^l 相互独立。在这样的模型中,设 X_{ij}^l 符合如下的概率分布:

$$\Pr(X_{ij}^l = 1) = \frac{w_j}{w_i + w_j} \quad (5.1)$$

$$\Pr(X_{ij}^l = 0) = \frac{w_i}{w_i + w_j} \quad (5.2)$$

5.2.2 全局排序的随机游走模型

考虑服务排序过程中的所有比较关系,可以绘制一个有向图 $G = (V, E)$ 对其进行描述。图中,节点集 V 代表被排序的服务集合,边集合 E 代表比较关系。将每条边表示为一个三元组 $\langle i, j, q_{ij} \rangle$, 其中 i 为起点, j 为终点, q_{ij} 表示 j 对 i 胜出的概率。

在实际比较评价的过程中,可以通过统计 j 排名高于 i 的次数而计算 q_{ij} 的值,即

$$q_{ij} = \begin{cases} \frac{1}{k} \sum_{l=1}^k X_{ij}^l & k \neq 0 \\ 0 & k = 0 \end{cases} \quad (5.3)$$

式中, k 是服务 (i,j) 总共进行比较的次数。若二者未进行比较,则令 $q_{ij} = q_{ji} = 0$ 。由于 X_{ij}^l 是相互独立的随机变量,由强大数定律(Strong Law of Large Numbers, SLLN)可知,当比较的数目足够多时(即 $k \rightarrow \infty$), q_{ij} 将收敛于 X_{ij}^l 的期望值,即

$$\lim_{k \rightarrow \infty} q_{ij} = \mathbf{E}[X_{ij}^l] = \frac{w_j}{w_i + w_j} \quad (5.4)$$

这样,可以将服务之间的比较过程形式化为一个图 G 上的随机游走过程。在每一步,从位置 i 跳至另一位置 j 遵循 p_{ij} 的概率分布。设在总的服务集合中,每两个服务之间的比较是随机选择的,因而 p_{ij} 定义为^[94]

$$p_{ij} = \begin{cases} \frac{1}{d_{\max}} q_{ij} & i \neq j \\ 1 - \frac{1}{d_{\max}} \sum_{k \neq i} q_{ik} & i = j \end{cases} \quad (5.5)$$

式中, d_{\max} 是图 G 中节点最大的出度。为了保证模型的收敛条件 $\sum_j p_{ij} = 1$, 这里增加了每个节点的自环。

在这样的模型设定中,由于随机游走中的每次变迁概率都与服务胜出的概率成正比,因此在大量比较(即稳态)的情况下,每个状态的访问概率(稳态概率)就反映了该状

态所代表服务的权值。如果对所有状态的稳态概率进行排序，即可得到所有服务的排序结果。

5.2.3 随机游走的马尔可夫模型

令 S_t 为时刻 t 该随机游走过程所处的状态，设 $X_t = S_{t+1} - S_t$ ，考虑到比较过程是独立同分布过程，因此对于任意整数 b, b_0, b_1, \dots, b_m ，有：

$$\begin{aligned} & \Pr(S_{m+n} = b \mid S_0 = b_0, S_1 = b_1, \dots, S_m = b_m) \\ &= \Pr\left(\sum_{t=m+1}^{m+n} X_t = b - b_m\right) \\ &= \Pr(S_{m+n} = b \mid S_m = b_m) \end{aligned} \quad (5.6)$$

此外，对任意 $i \in S$ ，有 $\sum_j p_{ij} = 1$ 。因此可以证明，该随机游走过程具有马尔可夫特性，可以视为一个离散时间马尔可夫链（DTMC）。该 DTMC 的转移概率矩阵用 \mathbf{P} 表示，其矩阵元素记为 $P_{ij} = \Pr(S_{t+1} = j \mid S_t = i) = p_{ij}$ 。

通过联立式 (5.7) 和式 (5.8)，可以求解得到 DTMC 的稳态概率向量 $\pi = [\pi_1, \pi_2, \dots, \pi_n]$ 。

$$\pi \cdot \mathbf{P} = \pi \quad (5.7)$$

$$\sum_{i=1}^n \pi_i = 1 \quad (5.8)$$

式 (5.7) 可以变形为 $\pi^T = \mathbf{P}^T \pi^T$ 形式。因此，当 \mathbf{P} 为常规转移矩阵时，如下两个结论成立。

- (1) 1 是矩阵 \mathbf{P}^T 的特征值；
- (2) 存在向量 $\vec{x} = [x_1, x_2, \dots, x_n]^T$ 是 1 所对应的特征向量。

这两个结论可以在理论上保证，可以通过求解矩阵 \mathbf{P}^T 对应特征值 1 的特征向量来求解马尔可夫链的稳态概率。考虑到式 (5.8) 所示的归一化条件，可以用下式计算 DTMC 的稳态概率：

$$\pi = \vec{x}^T / \sum_{i=1}^n x_i \quad (5.9)$$

5.3 协同服务评价算法

本节将讨论在大规模服务排名过程中，模型的参数获取方法，以及模型的求解算法。

5.3.1 算法概述

本节将提出排序聚合算法，旨在获得每个测试端对部分服务集合的排名后给出全局排序的结果。由于采用概率模型对比较排序过程进行描述，因而基于该模型的算法具有一定的鲁棒性，可以应对排序中可能存在的误差或噪声。这样，即使有少部分测试端使用了不同于其他大多数的排序标准，该算法仍能给出符合大多数测试端排序标准的全局排序结果。

基于比较模型，设计的全局排序聚合算法如下所述。

算法 5.1 全局排序聚合算法

1. 在测试端，将排序结果转化为两两比较的形式；
2. 将所有测试端的比较结果进行聚合，利用式 (5.5) 得到 DTMC 模型的转移概率矩阵 P ；
3. 对 DTMC 模型进行求解，得到状态的稳态概率 π ；
4. 根据稳态概率 π ，对所有服务进行排序，得到全局排序结果。

5.3.2 比较聚合方法

在算法 5.1 的第 1 步中，每个测试端需要将其维护的局部排序结果转化为两两比较的形式。设在测试端 j ，所有被排序的服务构成集合 S_j 。对于集合中的每个服务 $s \in S_j$ ， s 在集合 S_j 中的排名为 $r_j(s)$ ，则测试端 j 的局部排序可以由下述公式转化为两两比较对的集合形式：

$$C_j = \{(s_x, s_y) \mid r_j(s_x) > r_j(s_y), s_x \in S_j, s_y \in S_j\} \quad (5.10)$$

在算法的第 2 步中，转化完毕后，测试端将结果上传，并在中心节点进行聚合。中心服务器利用这些结果计算 5.2 节所述的 DTMC 模型参数。主要的参数可以由式 (5.11) 和式 (5.12) 计算得到。

$$q_{ij} = \frac{\sum_{k=1}^m |\{(i, j) \mid (i, j) \in C_k\}|}{\sum_{k=1}^m |\{(i, j) \mid (i, j) \in C_k \text{ or } (j, i) \in C_k\}|} \quad (5.11)$$

$$d_{\max} = \max_{i \in S} \left| \bigcup_{k=1}^m \{(i, j) \mid (i, j) \in S_k\} \right| \quad (5.12)$$

这里需要指出的是，在测试端将排序转化为两两比较的过程在一定程度上增加了通信的开销。若直接传递排序结果，则仅需要 $|S_j|$ 个服务排序结果即可，而转化后通信开销增长至 $(|S_j| - 1) |S_j| / 2$ ，增加了 $(|S_j| - 1) / 2$ 倍。但是，这一操作使得局部排序到两两比较的转化过程可以分布式地在每个测试端同时执行，从而将集中控制器上的任务下发至测试端，实现了并行化，理论上将计算时间开销缩减至集中处理的 $1/m$ 。在大规模服务环境中，当测试端数目较多时，即 $m \gg 1$ ，该并行化过程可以大幅减小计算成本。尤其是考虑到每个测试端是进行部分排序的服务，可以有 $|S_j| \ll |S| = n$ ，因此增加的额外通信开销倍数 $(|S_j| - 1) / 2$ 相比较节省的计算量倍数 m 是可以忽略的。综上所述，在测试端分布式地进行排序结果的转化，对算法效率的提高具有很大的促进作用。

5.3.3 求解马尔可夫链算法

在得到模型参数后，算法对 DTMC 模型进行求解，得到状态的稳态概率。当服务规模较少时，可以直接求解式 (5.7) 和 (5.8) 所示的平衡方程，从而得到稳态概率 π 。然而，当服务数目很多时，直接求解平衡方程需要大量的计算和存储开销，在实践中可能不再可行。因此，本节介绍两种工程实践中进行离散时间马尔可夫链求解的算法，以提高计算的效率。

1. 特征向量算法

在 5.2.3 节中, 已经证明可以通过计算对应特征值为 1 的特征向量计算 DTMC 的稳态概率。因此, 可以应用计算矩阵特征值和特征向量的算法, 随后执行式 (5.9) 得到稳态概率。这样, 算法 5.1 的第 3 步即可转化为数值分析中的特征向量计算这一经典问题。

这里可以使用数值分析领域中已有的一些特征向量算法对模型进行求解。例如, 幂迭代算法 (Power Iteration) 是目前广泛使用的一种循环型特征值求解方法。该算法的核心思路是, 利用特征值的定义不断进行乘法迭代。首先, 随机选取一个向量 \mathbf{b}_0 , 然后不断迭代执行式 (5.13), 直至向量 \mathbf{b}_k 收敛。

$$\mathbf{b}_{k+1} = \frac{\mathbf{P}^T \mathbf{b}_k}{\|\mathbf{P}^T \mathbf{b}_k\|_1} \quad (5.13)$$

在每次迭代中, 向量 \mathbf{b}_k 不断被矩阵 \mathbf{P}^T 左乘, 得到乘积后由乘积的 L_1 -范数进行归一化, 以保证算法的收敛性。这里, L_1 -范数又称为曼哈顿范数 (Manhattan Norm), 向量 $\mathbf{x} = [x_1, x_2, \dots, x_n]$ 的 L_1 -范数是该向量所有维分量的绝对值之和, 即 $\|\mathbf{x}\|_1 = \sum_i |x_i|$ 。

逆迭代算法 (Inverse Iteration) 是求解矩阵特征值和特征向量的另一常用方法。在本章讨论的问题中, 可以在矩阵 $(\mathbf{P}^T - \mathbf{I})^{-1}$ 上进行幂迭代, 获得矩阵的特征向量^[95]。此外, 近年来也有一些新的特征向量计算算法不断被提出。例如, 可以在逆矩阵上使用 MRRR (Multiple Relatively Robust Representations) 算法^[96]计算特征向量。

2. 迭代算法

由式 (5.7) 可知, 可以用迭代的方法直接计算稳态概率 π 。尤其是当转移概率矩阵 \mathbf{P} 中存在误差时, 利用迭代方法可以更好地得到稳态概率计算结果。令 π_t 表示在迭代第 t 步 DTMC 的概率分布, π_0 为任意选取的初始状态向量, 且保证 $\|\pi_0\|_1 = 1$ 。根据马尔可夫链固有的性质, 有如下的迭代公式成立:

$$\pi_{t+1} = \pi_t \cdot \mathbf{P} \quad (5.14)$$

算法 5.2 求解 DTMC 的迭代算法

1. 随机选取初始状态向量 π_0 , 满足 $\|\pi_0\|_1 = 1$;
 2. **for** $h \leftarrow 1$ **to** ∞ **do**
 3. $\pi_{h+1} \leftarrow \pi_h \mathbf{P}$;
 4. **if** $\|\pi_{h+1} - \pi_h\|_2 < \varepsilon$ **then**
 5. **return** π_{h+1} ;
 6. **end if**
 7. **end for**
-

经过足够多次的迭代后, 马尔可夫链将收敛至稳定状态, 即有 $\pi = \lim_{t \rightarrow \infty} \pi_t$ 。利用这样的性质, 可以用算法 5.2 迭代计算 DTMC 的稳态概率。其中, 向量 $\mathbf{x} = [x_1, x_2, \dots, x_n]$ 的 L_2 -范数 (又称为欧几里德范数) 的定义为 $\|\mathbf{x}\|_2 = \left(\sum_i x_i^2 \right)^{1/2}$ 。

5.4 算法分析

5.4.1 误差分析

首先,对算法的误差进行简要的分析。假设 \mathbf{P} 是模型中 DTMC 的真实的转移概率矩阵, $\tilde{\mathbf{P}}$ 是在排序聚合过程中观测得到的结果,那么观测的误差即为 $\mathbf{E} = \tilde{\mathbf{P}} - \mathbf{P}$ 。设 π 是转移概率矩阵为 \mathbf{P} 的 DTMC 的稳态概率,对应地, $\tilde{\pi}$ 是 $\tilde{\mathbf{P}}$ 对应的 DTMC 的稳态概率,根据稳态平衡方程,有 $\pi\mathbf{P} = \pi$ 和 $\tilde{\pi}\tilde{\mathbf{P}} = \tilde{\pi}$ 。另外,各状态的稳态概率之和为 1,即有 $\pi\mathbf{1} = \tilde{\pi}\mathbf{1} = 1$ 。根据以上等式,经过矩阵计算,有:

$$(\tilde{\pi} - \pi)(\mathbf{I} - \mathbf{P} + \mathbf{1}\pi) = \tilde{\pi}\tilde{\mathbf{P}} - \tilde{\pi}\mathbf{P} = \tilde{\pi}\mathbf{E} \quad (5.15)$$

令 $\mathbf{A} = \mathbf{I} - \mathbf{P}$, $\mathbf{A}^\#$ 表示矩阵 \mathbf{A} 的群逆矩阵 (Group Inverse), 在 DTMC 模型中, 有 $\mathbf{A}^\# = \{\mathbf{a}_{ij}^\#\} = (\mathbf{I} - \mathbf{P} + \mathbf{1}\pi)^{-1}$ [97]。于是, 由式 (5.15), 有 [98]:

$$\tilde{\pi} - \pi = \tilde{\pi}\mathbf{E}\mathbf{A}^\# \quad (5.16)$$

对式 (5.16) 中的每一维分量进行分析, 有:

$$|\tilde{\pi}_j - \pi_j| = \left| \sum_s \sum_t \tilde{\pi}_s e_{st} a_{jt}^\# \right| \leq \left(\sum_s \tilde{\pi}_s \right) \left(\max_r \sum_s |e_{rs}| \right) \left(\max_s |a_{sj}^\#| \right) \quad (5.17)$$

因此, 可以得到如下结论:

$$\max_j |\tilde{\pi}_j - \pi_j| \leq \|\mathbf{E}\|_1 \max_{i,j} |a_{ij}^\#| \quad (5.18)$$

综上所述, 式 (5.17) 和式 (5.18) 显示了存在测量偏差的情况下算法结果所呈现的最坏情况的误差。进一步可以证明, 采用 DTMC 模型进行的算法分析可以在测量过程中将误差限制在一定的界限内, 从而在一定程度上保证算法的性能。

5.4.2 时间复杂度分析

设整个评价框架中共有 n 个服务、 m 个测试端, 下面对整个算法的时间复杂度进行理论分析。

在 5.3.2 节中介绍的将局部排序转化至比较的算法, 需要将每个测试端得到的排序结果进行两两比较, 因而需要对所测试的服务集合进行两层嵌套迭代。因此在测试端 j , 该转化算法的时间复杂度为 $O(|S_j|^2)$ 。由于每个测试端可以并行执行该过程, 因而总体算法的时间复杂度为 $O((\max_j |S_j|)^2)$ 。在该过程后, 中心控制节点将所有结果进行聚合, 在最差情况下, 比较聚合过程可以保证在 $O(n^2)$ 时间内完成。

在 5.3.3 节中介绍了多种求解离散时间马尔可夫链的算法。其中, 大多算法是迭代求解, 因而很难给出具体确定性的时间复杂度分析。设其中某个迭代算法在 c 步迭代后收敛, 则 DTMC 求解算法的时间复杂度可以表示为 $O(cn^2)$ 。进一步, 在一些情况下, 步数 c 通常很小, 如已有研究证明, MRRR 算法可以在 $O(n^2)$ 时间内完成 [96]。

在得到稳态概率的基础上, 只需要对这些概率进行排序即可。 n 个服务的排序过程时间复杂度为 $O(n \log n)$ 。因此, 整体评价算法流程的时间复杂度为这些子过程的相加, 即 $O(n^2)$ 。

如果采用最朴素的办法在 m 个测试端分别对所有服务进行评价和排序, 该过程也需要 $O(mn)$ 的时间开销。可见, 本章提出的算法仅在时间复杂度上多花费了 $\frac{n}{m}$ 倍的开销, 却得到了如下两点显著的收益。

(1) 不必在测试端对所有的服务进行评价, 这一评价过程相比数学计算往往需要耗费大量的时间和资源开销, 尤其基于测量的方法还会对运行的服务产生额外的负载, 影响服务的性能, 进而造成评价结果的不准确, 因而这里提出的算法可以显著降低服务评价的成本。

(2) 该算法依据概率模型构建, 并考虑了观测中可能存在的误差及信息缺失, 最终得到的全局排序综合考虑了每个测试端的排序结果, 对误差和信息缺失具有很好的鲁棒性。

因此可以认为, 基于部分排序进行聚合得到全局排序的方法是对服务评价技术的发展和改进。

5.5 案例分析

本节将从实验验证的角度对所提出的算法的有效性进行讨论。本节基于实际存在的大规模服务 QoS 数据进行实验, 设定不同的信息缺失场景, 对算法的结果进行分析。

5.5.1 数据集简介

为了保证分析结果的真实有效性, 本节采用互联网上实际存在的大规模服务 QoS 测试数据进行案例分析。这里, 选取了 WSDream 数据集^[52,99]。该数据集是目前已公开的最大规模的 Web 服务 QoS 数据集, 包括 30 个国家 339 个测试端对互联网上的 5825 个 Web 服务进行测试的结果。测试过程中涵盖了近 200 万次的服务调用和评价过程, 得到了超过 390 万条测试数据。

在该数据集中, 每个 Web 服务共有两个 QoS 属性被评价和记录, 分别为响应时间和吞吐率。在实验中, 为了进一步显示算法的性能, 除这两个属性外, 引入了另一个人为规定的指标, 即 Power 公式^[56]。在第 2 章已经介绍过, Power 公式是衡量响应时间和吞吐率综合表现的评价标准, 也反映了这两个属性之间的关系。这里引入 Power 公式, 旨在除响应时间和吞吐率两个服务固有的属性外, 对这一人为规定的多属性综合评价函数也进行排序, 以显示算法对于不同的排序评价标准的适用性和有效性。令参数 $\alpha = 1$, 作为实验中的一个评价指标和排序标准, 则

$$\text{Power} = \frac{\text{吞吐率}^\alpha}{\text{响应时间}} = \frac{\text{吞吐率}}{\text{响应时间}} \quad (5.19)$$

在原数据集中, 每个测试端基本上对所有服务都进行了评价, 将它们的评价结果进行聚合, 可以得到全局的排序结果, 再将其作为排序的真实值 (Ground Truth), 即评价算法有效性的基准。随后, 在每个测试端按照一定比率随机选取部分服务进行评价, 作为实验的测试集。在测试集上应用本章提出的聚合算法, 然后与真实值进行比较, 从而评价算法的性能。

5.5.2 评价指标

为了直观显示算法的性能,评价算法得到的结果与真实值的差异,这里采用了如下3种评价指标。

1. 误差率

考虑到本章所建立的模型基于服务的两两比较,因此对得到结果所呈现出的两两比较情况进行评价。在评价过程中,将得到的全局排序转化为服务的两两比较对,然后与真实值进行对比,统计比较关系出现错误的数目。设 N_{err} 表示比较中发生了颠倒的数目, N_{comp} 表示比较的总个数,即 $N_{\text{comp}} = n(n-1)/2$, 则误差率 (Error Rate) 可以表示为

$$\text{Error Rate} = \frac{N_{\text{err}}}{N_{\text{comp}}} \quad (5.20)$$

2. 均方差

均方差 (Mean Squared Error, MSE) 是统计推断中常用的评价指标,用于对得到的估计值的准确性进行分析。在实验中,对排序结果和真实结果之间的差异进行分析,得到 MSE 的计算公式,即

$$\text{MSE} = \mathbf{E}_s [\hat{r}(s) - r(s)]^2 = \frac{1}{n} \sum_{i=1}^n (\hat{r}(s_i) - r(s_i))^2 \quad (5.21)$$

式中, $\hat{r}(s)$ 表示服务 s 在算法结果中的排序值, $r(s)$ 表示该服务排序的真实值。

3. 误差的概率密度函数

为了直观显示整体误差的分布情况,将讨论排序误差的概率密度函数 (Probability Density Function, PDF)。在实验中,将对排序误差 $\hat{r}(s) - r(s)$ 的值进行统计,绘制其 PDF 曲线,以显示排序误差的分布情况。

5.5.3 实验结果

设置每个测试端随机删除 20% ~ 80% 的测试数据,分别进行实验,以对算法性能进行评价。例如,缺失 20% 数据代表测试端仅对原有的 80% 的服务进行评价,删除其余 20% 的服务,然后对这 80% 的服务按照 QoS 值进行排序。实验分别将响应时间、吞吐率和 Power 值作为排序的标准,变化缺失数据的比率,得到实验排序结果,对其进行 3 个评价指标的分析,以显示算法的正确性。

图 5.2 所示的是基于响应时间排序标准的实验结果。图 5.2 (a) 反映了不同的信息缺失率对比较排序结果造成的影响。从实验结果中可以看出,随着缺失信息的增多 (即每个测试端评价服务数目的减少),比较排序的误差逐渐增大。但是,从实验数据可见,即使缺失了 80% 的测试数据,本章所提出算法的误差率仍可保证在 2.4% 以内,算法的有效性可见一斑。图 5.2 (b) 反映了排序结果的均方差随信息缺失率变化的情况,与图 5.2 (a) 的结论一致,缺失率的增加将导致排序结果误差的增大。在 80% 缺失率的最坏情况下,均方差可以保证在 1.1×10^5 内。考虑到排序中共有 5825 个服务,均方差的最大值近似为 $5825^2 \approx 3.4 \times 10^7$ 。与实验结果对比可知,本章提出的算法可以将排序误差控制在一个较小的范围内。图 5.2 (c) 所示的是排序误差的 PDF 曲线。可见,信息缺失率的增加会导致绝对误差的增多,在 PDF 上显示为曲线的变宽,即排序误差方差的增加。从误差分布可以看出,绝

大多数服务排序的绝对值误差均在 0 附近。该算法可以保证以很高的概率将绝对误差限制在 $-500 \sim +500$ 的范围内 ($500/5825 \approx 8.6\%$)。

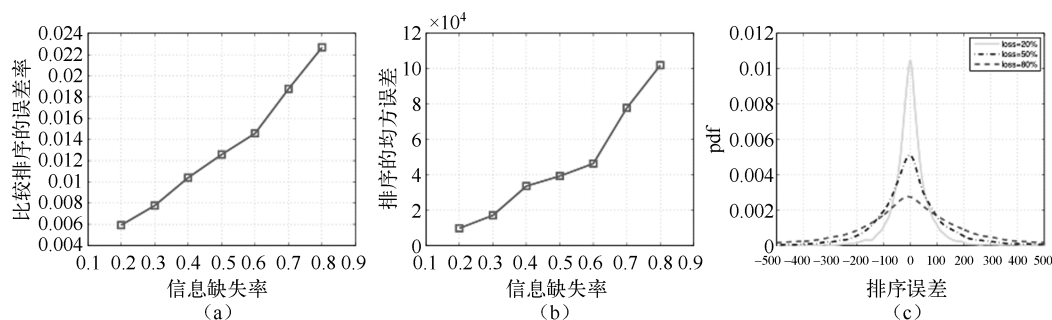


图 5.2 基于响应时间排序标准的实验结果

图 5.3 和图 5.4 所示的是基于吞吐率和 Power 值排序标准的实验结果。从实验数据中可以得到与前述响应时间排序标准实验相一致的结论。这些实验结果说明，本章所提出的算法可以有效应用于不同的 QoS 评价标准，具有普适性。另外，实验结果可以证明，该算法可以有效地应对排序过程中的信息缺失情况，能够得到符合服务 QoS 内在规律的排序评价结果。因此，利用本章提出的技术，测试端可以仅对一小部分服务进行评价，从而可以显著地降低评价开销，并同时保证排序评价结果的准确性。

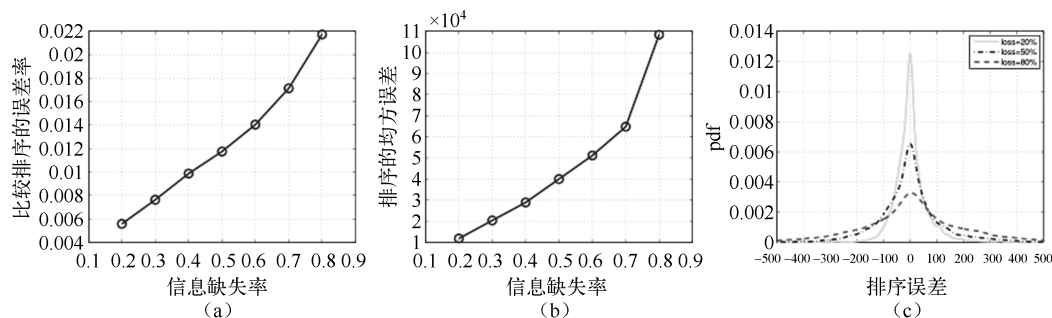


图 5.3 基于吞吐率时间排序标准的实验结果

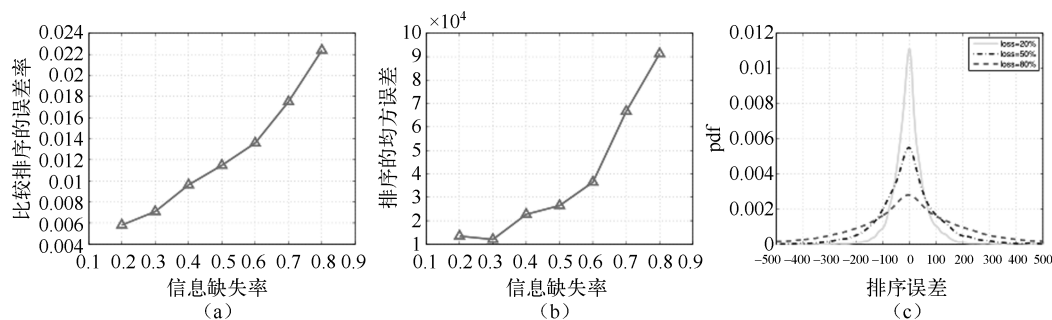


图 5.4 基于 Power 公式排序标准的实验结果

5.6 本章小结

在大规模的服务计算背景下,服务的评价在有效性和效率上遇到了很大的挑战。本章针对服务选择、服务推荐等过程中的服务排序问题,研究了有效降低服务评价成本、同时保证排序准确性的方法和技术。本章建立了服务比较的概率模型,用于描述服务之间的比较关系,进而将服务排序形式化为随机游走过程,并利用离散时间马尔可夫模型进行求解。在该模型基础上,提出了服务排序聚合方法,给出了从分布式的部分排序整合为全局排序的算法,并在理论上给出了算法的误差分析和复杂度分析。最后,在大规模服务 QoS 数据集上进行了模拟实验,验证了算法的有效性。本章提出的机制能够在服务评价的过程中有效降低评价的成本,并且能够容忍观测中可能存在的误差,在服务的评价在理论分析和技术方法上具有一定的推进作用。

基于折扣马尔可夫决策的性能— 能耗综合优化

近年来,能耗效率问题成为服务计算中的一个研究热点。如何提供更好的服务,满足用户的服务质量需求,同时尽量减小系统的能耗,降低服务的运行成本,是其中一个重要的科学问题。

一个服务计算系统可以向用户提供诸多的功能,每个功能也可由诸多服务提供支持,这些服务具有相同或相似的功能,可以被用户或服务中间商选择。每个服务部署和运行在底层诸多的物理机上。在这样的系统中,可以有如下两种提高系统能耗效率的方法。

(1) 在服务层进行有效的服务选择。服务选择是服务计算中的一个经典问题,旨在从众多功能相同或相似的服务中选择一个或一组服务,在满足用户功能需求的情况下,最大化服务质量,同时最小化运营成本^[100]。考虑到可选服务分布在众多物理机上,可以根据用户的功能需求选择合适物理机上的合适的服务,尤其是可以将某些没有服务运行的服务器设置为睡眠状态,以达到更好的能耗效率^[101]。

(2) 在系统层进行动态速率控制。CPU 是服务器运行中消耗能源最多的部件^[102]。为了更高效地利用电力资源,当今主流的 CPU 均整合了动态调整主频和电压的技术,即动态电压和速率调节技术 (Dynamic Voltage and Frequency Scaling, DVFS)。例如,Intel 公司的 CPU 具有的 SpeedStep^[103]、TurboBoost^[104] 技术,AMD 公司的 CPU 所整合的 PowerNow!^[105] 和 Cool n Quiet^[106] 等技术。提供动态调整 CPU 的电压和主频的机制,可以使系统根据负载或需求情况灵活改变运行状态,令其在保证性能的条件下运行在合适的速率和电压下,以达到能耗的最优利用^[26]。

本章将综合考虑服务选择和速率控制,从服务层和系统层两个角度同时研究服务计算系统的能效优化问题。本章将建立服务计算系统的性能—能耗评价模型,给出模型分析方法和求解技术。在此基础上,提出分布式优化框架和相关算法,并通过理论证明和案例分析证明算法的有效性。

6.1 问题描述

本节从服务选择过程出发,结合动态速率控制,对服务计算系统中的能耗优化问题进行

形式化描述。

设用户的需求为一个包含 m 个任务的工作流，表示为 $I = \{1, 2, \dots, m\}$ 。这里为了进一步分析的便捷，假设这 m 个任务之间不存在循环、分支等复杂的依赖关系。系统中与用户请求相关的所有服务集合定义为 $J = \{1, 2, \dots, n\}$ 。设可以完成用户任务 $i \in I$ 的服务集合为

$J_i = \{j_1, j_2, \dots, j_{n_i}\} \subseteq J$ ，这 n_i 个服务部署在不同的服务器上。进而，有 $n = \sum_{i=1}^m n_i$ ， $J = \bigcup_{i=1}^m J_i$ 。

令 p 表示该系统中服务器的数目，服务器集合表示为 $K = \{1, 2, \dots, p\}$ 。将服务与服务器的对应关系用矩阵 $\mathbf{L}_{n \times p} = [l_{jk}]_{n \times p}$ 表示，矩阵中的每个元素 $l_{jk} \in \{0, 1\}$ 指示了服务 j 是否部署在服务器 k 上。这里不考虑服务的部署问题，即假设矩阵 \mathbf{L} 是常量矩阵。定义服务选择的决策变量为 $x_{ij} \in \{0, 1\}$ ，指示服务 j 是否被选择用于完成任务 i 。

本研究的目标是在整个服务计算系统中达到最好的能耗效率，即得到最大的利润。假设完成服务 i 可以得到收益 $v(i)$ ，它可以由 SLA 协议所决定。另一方面，服务器运行过程中需要一定的开销，将其定义为 $c(k)$ 。这里主要讨论系统长期运行中的能耗效率，因而假定系统开销主要与系统的能耗相关，忽略初次构建系统的成本。考虑到动态速率控制主要影响服务器的能耗状态，与服务器的开销直接相关，因此 $c(k)$ 可以在一定意义上视为能效优化问题中的另一个决策变量。在达到目标的同时，还应保证用户功能需求的满足，即对于工作流中的每个任务 $i \in I$ ，均保证有某个服务 $j \in J_i$ 被选择。综上所述，服务计算系统中服务选择和动态速率控制可以形式化地表示为如下优化问题：

$$\underset{x_{ij}, c(k)}{\text{maximize}} R = \sum_{i \in I} \sum_{j \in J_i} x_{ij} v(j) - \sum_{k \in K} c(k); \quad (6.1)$$

s. t.

$$\sum_{j \in J_i} x_{ij} = 1, \quad \forall i \in I; \quad (6.2)$$

$$x_{ij} \in \{0, 1\}, \quad \forall i \in I, \forall j \in J_i, \quad (6.3)$$

6.2 服务系统的性能—能耗评价模型

为了解决能效优化的问题，首先研究单个服务器的性能和能耗的分析评价。本节将建立服务器的马尔可夫回报模型（MRM），研究模型的分析求解方法，给出性能和能耗属性的量化分析数学表达。

6.2.1 原子服务的性能模型

从自上而下的角度出发，一个服务器可以模型化为一个有控制器的排队系统，该控制器在每个控制时间点对系统的速率进行决策，从有限的待选集合中选择最优的运行速率，达到最好的性能—能耗效率^[107]。每个服务器维护了各自的请求缓存队列，承载着各种服务，并且假设服务器之间相互独立。已有研究工作指出，会话层以上的任务到达服从泊松分布^[74]。考虑到泊松过程的可叠加性，因而每个服务器上的所有任务请求仍然服从泊松分布，并且总的到达速率是所承载所有服务请求的到达速率之和。令 λ_j 表示服务 j 的请求到达速率，则服务器 k 的任务请求到达速率为

$$\lambda = \sum_{i \in I} \sum_{j \in J_i} x_{ij} l_{jk} \lambda_j \quad (6.4)$$

假设任务大小和服务时间服从指数分布^[75]，则单个服务器的服务处理过程可以模型化为一个连续时间马尔可夫链（CTMC），如图 6.1 所示。

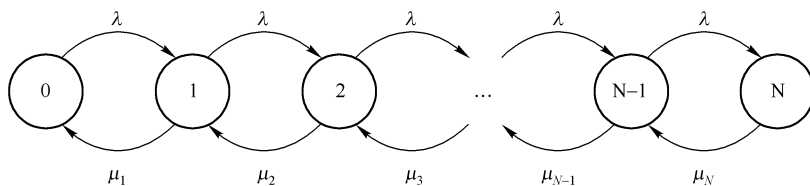


图 6.1 服务器的 CTMC 模型

该模型中的状态定义为服务器缓存队列中的服务数目； μ_i 代表状态 i 时的服务速率，该服务速率由控制器所决定； N 为缓存队列的容量。该 CTMC 的转移速率用 $Q_{N \times N} = [q_{i,j}]_{N \times N}$ 表示，其中，

$$q_{i-1,i} = \lambda = \sum_j \lambda_j \quad i = 1, 2, \dots, N \quad (6.5)$$

$$q_{i,i-1} = \mu_i \quad i = 1, 2, \dots, N \quad (6.6)$$

$$q_{i,i} = - \sum_{j \neq i} q_{i,j} \quad i = 0, 1, \dots, N \quad (6.7)$$

$$q_{i,j} = 0 \quad |i-j| \geq 2, \quad i, j = 0, 1, \dots, N \quad (6.8)$$

在模型参数基础上，下面给出模型的稳态概率分析。令 $\pi = [\pi_0, \pi_1, \dots, \pi_N]$ 表示状态的稳态概率向量，则它可以由如下方程组求解得到：

$$\begin{cases} \pi \cdot Q = 0 \\ \sum_{i=0}^N \pi_i = 1 \end{cases} \quad (6.9)$$

得到稳态概率后，可以得到服务器的利用率，即

$$\rho = \sum_{i=1}^N \pi_i = 1 - \pi_0 \quad (6.10)$$

服务器中的平均任务数为

$$N = \sum_{i=0}^N i \cdot \pi_i \quad (6.11)$$

因而利用 Little 公式，可以得到该服务器上请求的平均响应时间，即

$$T = \frac{N}{\lambda} = \frac{1}{\lambda} \sum_{i=0}^N i \cdot \pi_i \quad (6.12)$$

由于服务器的控制器单元在每个状态会调节系统运行主频和电压，导致服务速率的不同，因此服务器在每个状态完成服务的概率也不同。在一个时间段 τ 内，若服务速率为 μ_i ，则服务器可以完成一个服务的概率 p_{srv_i} 为

$$p_{\text{srv}_i} = \frac{\mu_i}{\mu_i + \lambda} \cdot (1 - e^{-(\lambda + \mu_i) \cdot \tau}) \quad (6.13)$$

那么，在这段时间内，该服务器上可以获得收益的期望值 v_i 就是服务收益的期望值 \bar{v} 乘以 p_{srv_i} 。其中， \bar{v} 和 v_i 的计算方法为

$$\bar{v} = \frac{\sum_{i \in I} \sum_{j \in J_i} x_{ij} l_{jk} v(j)}{\sum_{i \in I} \sum_{j \in J_i} x_{ij} l_{jk}} \quad (6.14)$$

$$v_i = \bar{v} \cdot p_{\text{sv}i} \quad (6.15)$$

除服务得到的收益外, 服务过程中服务器需要消耗能源, 因而需要对服务器的功率和能耗进行分析。在电子计算机系统中, 服务器的功率分为静态功率和动态功率两种, 前者由漏电流产生, 它独立于系统运行的主频和利用率; 后者由开关电路活动产生, 与时钟主频、利用率、I/O 活动密切相关^[26]。当今的 CPU 动态速率和电压调整技术可以根据系统负载和运行状态调整时钟主频, 进而决定服务速率。一般认为, CPU 主频与服务速率成正比。服务器的能耗可以由下式计算得到

$$P_i = P_s + P_{d_i} = P_s + \theta_f \cdot \mu_i^\alpha \quad (6.16)$$

式中, P_s 为静态功率; P_{d_i} 为动态功率; θ_f 和 α 为常数系数, 且有 $\alpha \geq 1$ 。

因此, 在时间段 τ 内, 系统能耗造成的运营开销等于系统的能耗乘以电价 \bar{c} , 即

$$c_i = \bar{c} \cdot P_i \cdot \tau \quad (6.17)$$

综合考虑系统的服务收益和能耗开销, 可知系统服务过程中在状态 i 的利润为

$$r_i = v_i - c_i \quad (6.18)$$

将 r_i 定义为状态 i 的回报值, 就得到了服务器的 MRM 模型。

在稳态意义上, 整个服务器可以获得的利润的期望值 (即 MRM 模型的稳态回报值) 可以通过下式计算得到:

$$\bar{R} = \sum_{i=0}^N \pi_i \cdot r_i \quad (6.19)$$

6.2.2 嵌入离散时间马尔可夫回报模型

由于 CTMC 的时间参数可以在正实数域任意取值, 因此它具有强大的模型描述能力, 在性能评价中可以发挥巨大的作用。但是, 计算机是一个离散时间事件系统, 其操作指令根据系统内部时钟周期性执行。为了使分析过程更符合系统特性, 进一步细化系统的行为特性, 并为下一步的优化决策奠定理论基础, 本节将已建立的 CTMC 嵌入其对应的 DTMC, 将连续时间回报模型转化为离散时间的模型。服务器的嵌入 DTMC 模型如图 6.2 所示。

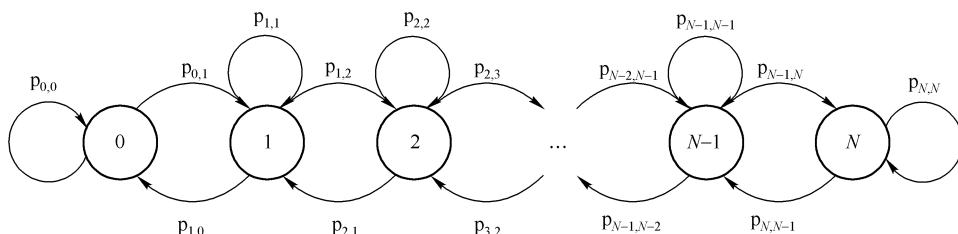


图 6.2 服务器的嵌入 DTMC 模型

在嵌入过程中, 假定每个相邻状态转移的时间间隔均为定值 τ , 即计算机根据固定的时钟发生器进行执行和决策, 则嵌入 DTMC 的转移概率矩阵 $\mathbf{P}_{N \times N} = [p_{i,j}]_{N \times N}$ 的计算公式为

$$p_{i,j} = \frac{q_{i,j}}{\sum_{k \neq i} q_{i,k}} \cdot (1 - e^{-\sum_{k \neq i} q_{i,k} \times \tau}) \quad i \neq j \quad (6.20)$$

$$p_{i,i} = e^{-\sum_{k \neq i} q_{i,k} \times \tau} \quad (6.21)$$

嵌入 DTMC 的稳态概率（即 $\nu = [\nu_0, \nu_1, \dots, \nu_N]$ ）可以通过求解如下方程组得到：

$$\begin{cases} \nu = \nu \cdot P \\ \sum_{i=0}^N \nu_i = 1 \end{cases} \quad (6.22)$$

另外，原 CTMC 的稳态概率也可通过嵌入 DTMC 计算得到：

$$\pi_i = \frac{\nu_i h_i}{\sum_{j=0}^N \nu_j h_j} = \frac{\nu_i}{\sum_{j=0}^N \nu_j} = \nu_i \quad (6.23)$$

由于嵌入时间间隔 $h_j = \tau$ ，因此可以证明，通过嵌入 DTMC 得到的稳态概率与原始的 CTMC 的稳态概率完全相等。从而可以证明，该模型嵌入过程在稳态分析上是正确、有效的。

在不变回报值定义的条件下，结合式（6.23）所得到的结论，可以进一步证明原 CTMC 的稳态回报值与嵌入 DTMC 的稳态回报值的等价性：

$$\overline{R'} = \sum_{i=0}^N \nu_i \cdot r_i = \sum_{i=0}^N \pi_i \cdot r_i = \overline{R} \quad (6.24)$$

6.2.3 模型分析的更新技术

在马尔可夫模型分析中，摄动分析（Perturbation Analysis）是其中一个重要的方面。它所关注和解决的问题是，当马尔可夫链中由某一个状态转出的概率有所变化时，模型分析结果的变化情况。在服务计算系统中，当控制器在某个状态 i 的速率控制策略变化时，就会影响该状态的转出概率。由于服务计算具有高动态性的特点，控制器会根据负载变化情况频繁调整控制策略，从而造成模型分析中的参数扰动。如何高效地进行摄动分析，避免每次调整后需要重新求解模型带来的巨大计算资源开销，是亟待解决的问题。本节将 Rank-1 Update 技术^[108]用于模型的分析求解，给出存在扰动的情况下高效的马尔可夫链更新方法。

假设在进行动态的服务速率控制时，状态 k 的速率从 μ_k 变为 μ'_k ，则嵌入 DTMC 的转移概率矩阵 P 变为 $P' = P + \Delta P$ 。其中， P 仅有第 k 行会发生变化，其他行保持不变。 $\Delta P = [\Delta p_{i,j}]$ 的秩为 1^[97]，其值为

$$\Delta p_{k,k-1} = \frac{\mu'_k}{\mu'_k + \lambda} \cdot (1 - e^{-(\mu'_k + \lambda)\tau}) - \frac{\mu_k}{\mu_k + \lambda} \cdot (1 - e^{-(\mu_k + \lambda)\tau}) \quad (6.25)$$

$$\Delta p_{k,k+1} = \frac{\lambda}{\mu'_k + \lambda} \cdot (1 - e^{-(\mu'_k + \lambda)\tau}) - \frac{\lambda}{\mu_k + \lambda} \cdot (1 - e^{-(\mu_k + \lambda)\tau}) \quad (6.26)$$

$$\Delta p_{k,k} = e^{-(\mu'_k + \lambda)\tau} - e^{-(\mu_k + \lambda)\tau} \quad (6.27)$$

根据 ΔP 的表达，它可以写为向量 $\mathbf{a} = [a_i]^T$ 和 $\mathbf{b} = [b_i]^T$ 的乘积的形式，即

$$\Delta P = \mathbf{a}\mathbf{b}^T$$

式中，

$$a_i = 1 \quad i = k \quad (6.28)$$

$$a_i = 0 \quad i \neq k \quad (6.29)$$

$$b_i = \Delta p_{k,i} \quad i = k-1, k, k+1 \quad (6.30)$$

$$b_i = 0 \quad \text{其他} \quad (6.31)$$

这样, 可以将 Sherman – Morrison 公式应用到矩阵 $A = I - P$ 上^[109]。这里需要先计算矩阵 A 的群逆 (Group Inverse) 矩阵 $A^\#$ 。它是满足 $AA^\#A = A$ 、 $A^\#AA^\# = A^\#$ 和 $AA^\# = A^\#A$ 的唯一矩阵^[110]。在本章的模型分析中, $A^\#$ 可以由下式得到^[97]:

$$A^\# = (I - P + \mathbf{1} \cdot \pi^T)^{-1} \quad (6.32)$$

由相关理论可知, 更新后的 DTMC 的稳态概率 π' 的计算公式为

$$\pi' = \pi \cdot \left(I + \frac{ab^T A^\#}{1 - b^T A^\# a} \right) \quad (6.33)$$

更新后的矩阵 $A' = I - P'$ 的群逆矩阵 $A'^\#$ 可由下式计算得到:

$$A'^\# = (I - \pi' a \mathbf{1} b^T A^\#) \left(A^\# + \frac{A^\# ab^T A^\#}{1 - b^T A^\# a} \right) \quad (6.34)$$

最后, 得到式 (6.33) 所示结果, 更新后的 DTMC 的回报值可以由式 (6.24) 计算得到。

从更新 DTMC 的计算过程可以看出, 该更新技术可以避免在更新后通过求解线性方程组完全重新计算每个状态的稳态概率, 仅需根据变化量结合群逆矩阵即可通过简单计算得到更新后的结果。在更新过程中, 群逆矩阵也可通过更新算法简化计算并不断迭代, 从而降低了模型分析的复杂度。在调整系统中服务器的速率控制策略时, 可以应用该技术以简化计算过程。此外, 该技术也可被应用至敏感度分析中, 用于研究转移概率参数对评价结果的影响程度。

6.3 基于马尔可夫决策的动态功耗控制

为了优化系统的能耗效率, 最大化收益, 需要得到最优的速率控制策略, 从而得到模型中回报值的最大值。本节将该优化过程形式化为一个马尔可夫决策过程, 然后进行分析求解。随后, 从理论上证明马尔可夫决策优化和所建立的 MRM 模型优化是等效的, 从而得到马尔可夫决策对回报值优化的有效性。

6.3.1 问题描述和求解方法

在每个状态下, 控制器需要确定最优的速率控制策略, 以达到收益最大化。因此, 可以将该优化过程形式化为一个马尔可夫决策过程 (Markov Decision Process, MDP)。该 MDP 的决策周期、状态空间、决策空间、回报函数和优化目标的定义如下所述。

1) 决策周期 每个决策周期的长度为 τ , 按照时间顺序编号为 $n \in \{1, 2, \dots\}$ 。对应地, 每次决策发生在时间 $t = \tau, 2\tau, \dots$ 。

2) 状态空间 在第 n 个决策周期的状态 $S(n)$ 定义为当前时间缓存队列中存在的任务数目, 即 $S(n) \in S = \{0, 1, \dots, N\}$ 。

3) 决策空间 第 n 个决策周期, 控制器需要从 k 个离散值中选取最优的服务速率。决策行为表示为 $a_n \in A = \{\mu_1, \mu_2, \dots, \mu_k\}$ 。为了进一步提高能源使用效率, 在每个状态, 服务器不仅可以运行在某个常数的服务速率下, 也可以进入睡眠状态, 暂停服务过程^[111]。因此, 令 $\mu_1 = 0$, 代表服务器进入睡眠的状态。

4) 回报函数 定义 MDP 中每个状态的回报函数 $r(S(n), a)$ 为 MRM 中定义的回报函数, 即 $r(S(n), a) = r_n$, 其中 r_n 为式 (6.18) 所示的 MRM 回报函数。

5) 优化目标 在决策周期 n , 定义 MDP 每个状态的优化目标为 $V_n(S(n))$, 代表了当前状态的回报值加折扣的未来回报的期望值。这里借鉴经济学的评价思路, 引入折扣因子

$\gamma \in (0,1)$ ，区分既得收益和未来收益；对未来收益乘以折扣因子，代表在该经济模型中，相比于具有不确定性的未来收益，更看重当前的既得收益。综上所述，优化目标 V_n 表达式为

$$V_n(S(n)) = \max_{a_n \in A} \left\{ r(S(n), a_n) + \gamma \sum_{s' \in S} p(s' | S(n), a_n) V_{n+1}(s') \right\} \quad (6.35)$$

对于该 MDP 模型的求解，可以使用已有的一些方法，如值迭代算法（Value Iteration）和策略迭代算法（Policy Iteration）^[112]。这两个算法如算法 6.1 和算法 6.2 所示。可以证明，这两个算法在求解效果上是等效的，它们在充分迭代后均收敛至最优解^[112]。在值迭代算法中，每次迭代需要的计算次数为 $\sum_{s \in S} |A| \times |S| = k(N+1)^2$ ；而每次策略迭代需要计算次数是 $\sum_{s \in S} |A| \times |S| + \frac{1}{3} |S|^3 = k(N+1)^2 + \frac{1}{3}(N+1)^3$ 。虽然策略迭代每次需要的计算复杂度偏高，但通常该算法需要的迭代次数较少。

算法 6.1 值迭代算法

```

1: 选取  $V_0$ ,  $\varepsilon > 0$ ;
2: for  $n \leftarrow 0$  to  $\infty$  do
3:   for all  $s \in S$  do
4:      $V_{n+1}(s) \leftarrow \max_{a \in A} \left\{ r(s, a) + \gamma \sum_{s' \in S} p(s' | s, a) \cdot V_n(s') \right\}$ ;
5:   end for
6:   if  $\max_{s \in S} \{ |V_{n+1}(s) - V_n(s)| \} < \varepsilon$  then
7:      $a_\varepsilon(s) \leftarrow \arg \max_{a \in A} \left\{ r(s, a) + \gamma \sum_{s' \in S} p(s' | s, a) \cdot V_{n+1}(s') \right\}$ ;
8:   return  $a_\varepsilon$ ,  $V_{n+1}$ ;
9:   end if
10: end for
```

算法 6.2 策略迭代算法

```

1: 选取任意策略  $a_0 \in A$ ;
2: for  $n \leftarrow 0$  to  $\infty$  do
3:   求解方程  $(I - \gamma P_{a_n}) V_n = r_{a_n}$ , 得到  $V_n$ ;
4:   选取最优策略  $a_{n+1} \leftarrow \arg \max_{a \in A} \{ r_a + \gamma P_a V_n \}$ ;
5:   if  $a_{n+1} = a_n$  then
6:      $a^* \leftarrow a_n$ ;
7:     return  $a^*$ ,  $V_n$ ;
8:   end if
9: end for
```

另外, 可以将 DTMC 求解的更新技术应用在策略迭代算法的第 4 步中, 从而将每次迭代的计算复杂度进一步缩减至 $3k(N+1) + \frac{1}{3}(N+1)^3$ 。

6.3.2 马尔可夫决策对马尔可夫回报优化的有效性证明

本节证明 MDP 优化对 MRM 回报值优化的有效性。提出了定理 6.1, 并通过对 MDP 回报值和 MRM 回报值的对应关系分析, 给出该定理的理论证明。

【定理 6.1】 MDP 优化算法可以得到 MRM 模型的最优回报值。

证明: 相关研究证明, 算法 6.1 和算法 6.2 执行完成 (收敛) 后, 可以得到 MDP 回报值最优, 即式 (6.35) 中的 $V_n(S(n))$ 取最大值^[112]。在稳态分析中, 当算法收敛 (即 $n \rightarrow \infty$) 时, 式 (6.35) 满足如下等式:

$$V_i = r_i + \gamma \sum_j V_j \cdot p_{i,j} \quad (6.36)$$

令 $\mathbf{V} = [V_0, V_1, \dots, V_N]^T$, $\mathbf{R} = [r_0, r_1, \dots, r_N]^T$, 得到式 (6.35) 的矩阵表示, 即

$$\mathbf{V} = \mathbf{R} + \gamma \mathbf{P}\mathbf{V} \quad (6.37)$$

上式两边左乘 CTMC 的稳态概率 $\pi = [\pi_0, \pi_1, \dots, \pi_N]$, 有:

$$\pi \mathbf{V} = \pi \mathbf{R} + \gamma \pi \mathbf{P}\mathbf{V} \quad (6.38)$$

稳态时, CTMC 和 DTMC 的马尔可夫回报模型参数具有如下关系:

$$\pi = \nu = \nu \cdot \mathbf{P} = \pi \cdot \mathbf{P} \quad (6.39)$$

该服务器的 MRM 稳态回报值有如下矩阵形式的表达:

$$\bar{\mathbf{R}} = \pi \cdot \mathbf{R} \quad (6.40)$$

结合式 (6.38)、式 (6.39) 和式 (6.40), 得到:

$$\pi \mathbf{V} = \bar{\mathbf{R}} + \gamma \pi \mathbf{P}\mathbf{V} \quad (6.41)$$

对上式进行变形, 可得到如下结论:

$$(1 - \gamma) \pi \mathbf{V} = \bar{\mathbf{R}} \quad (6.42)$$

由以上对应关系可知, 当 MDP 算法收敛 (即 \mathbf{V} 优化) 后, MRM 模型的稳态回报值 $\bar{\mathbf{R}}$ 也可完成优化过程, 且两个模型之间的回报值符合式 (6.42) 所示的相互关系。从而定理 6.1 得证。

6.4 分布式的动态服务选择与功耗控制综合优化

在单服务器的模型、分析和优化基础上, 考虑到服务的动态性特点, 本节从上层服务的角度扩展该研究问题, 综合考虑服务层的服务组合及系统层的速率控制, 全面优化系统的能耗效率, 以达到最优的收益。

6.4.1 优化框架

Agent 是一个具有自适应性和智能性的计算机软件实体, 它能以主动服务的方式进行决策, 完成用户指定的任务, 满足用户的需求^[113]。多个相互链接的 Agent 可以组成多 Agent

系统，它们可以相互协作、协同、协商，更高效地完成更复杂的任务。目前，多 Agent 技术已广泛应用于 Web 服务、SOA、云计算等服务计算系统，取得了良好的效益^[114-116]。本节将多 Agent 技术应用至服务计算系统中的能效优化，综合考虑服务层和系统层，协同分布式地进行高效的动态优化。

基于多 Agent 技术的服务选择和服务器速率控制的综合优化框架如图 6.3 所示。

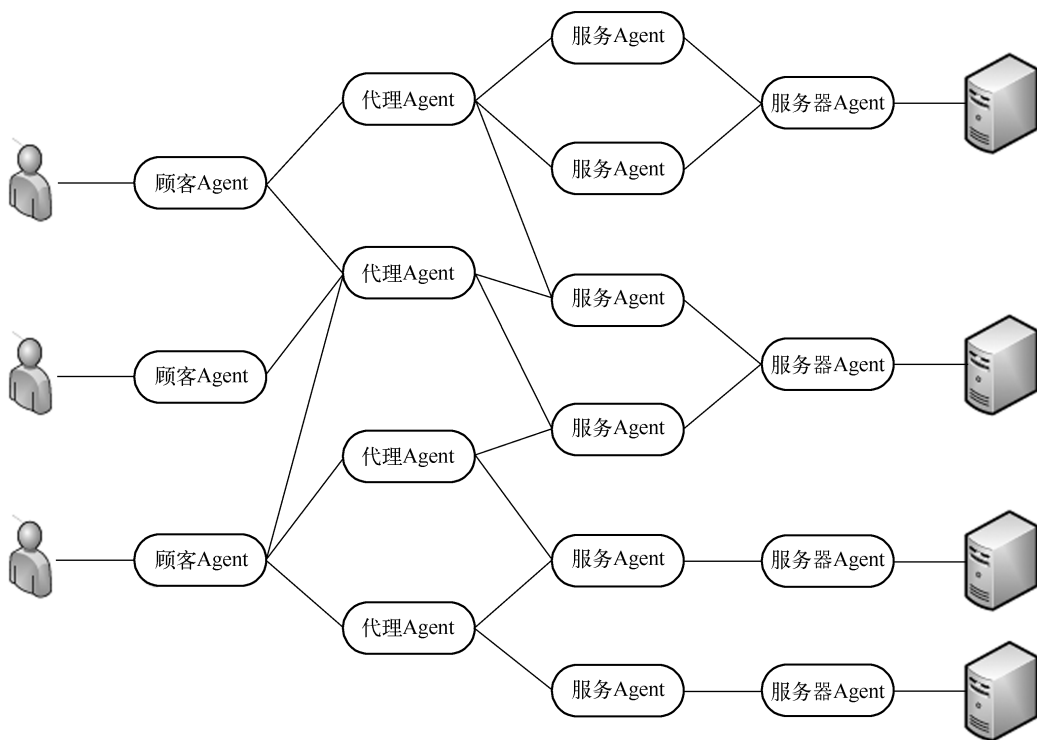


图 6.3 基于多 Agent 技术的服务选择和服务器速率控制的综合优化框架

该框架中共有 4 类 Agent，分别是顾客 Agent、代理 Agent、服务 Agent 和服务器 Agent。其中，顾客 Agent 直接与用户进行交互，接受用户的任务需求，进行整合后将任务分发给代理 Agent；代理 Agent 是服务选择过程中的核心部件，每个代理 Agent 根据用户需求及自身所维护的信息，从其管理的服务 Agent 中选择最优服务，组合成为服务工作流，提供给上层的顾客 Agent；服务 Agent 主要负责服务的管理，包括监控和服务质量信息的维护，它是代理 Agent 和服务器 Agent 之间的桥梁，将上层服务与底层硬件联系起来；服务器 Agent 负责物理服务器的软、硬件管理，为服务的执行提供支撑，并从系统层执行速率控制策略，以达到能效最优。

6.4.2 优化算法

在该服务选择和服务器速率控制的综合优化框架中，每层 Agent 各自负责不同的功能。基于它们各自管理的有限的信息和资源，各层 Agent 可以找到各自所负责问题的局部最优解。最上层的 Agent（即顾客 Agent）可以将所有相关信息进行整合，从局部最优解得到全

局最优解。同一层的 Agent 可以并行地执行任务，将一个复杂的优化问题分割为多个优化子问题分别求解，从而显著提高计算效率。这里，借鉴经济学中的边际效用的优化思路，进行优化问题的求解。

首先自下而上地讨论服务器 Agent 的优化策略。当新的任务请求即将到达时，服务器 Agent 首先尝试将新的请求接纳执行，给出新的最优值的预测。具体地，首先服务 Agent 根据当前的系统参数和负载情况，应用 MDP 算法求解得到新的最优能耗控制策略。随后，将原有的 MDP 最优回报值与新的最优回报值进行比较，将其边际效益（回报值差值）返回给上层的服务 Agent。该过程称为局部优化算法，如算法 6.3 所示。

算法 6.3 局部优化算法

输入： λ'

输出： ΔV

局部变量： $\lambda, \lambda_{\text{new}}, a, a_{\text{new}}, V, V_{\text{new}}$

1. $\lambda_{\text{new}} \leftarrow \lambda + \lambda'$;
 2. 在新的参数 λ_{new} 下，使用值迭代算法（算法 6.1）或策略迭代算法（算法 6.2），找到新的最优策略 a_{new} 和最优回报值 V_{new} ;
 - 3: **return** $\Delta V \leftarrow V - V_{\text{new}}$;
-

当上层的 Agent 根据其掌握的局部信息最终决定将新的任务请求分发至该服务器上时，服务器 Agent 将更新所有系统参数，并将已在局部优化算法中计算好的最优值作为新的回报值。这个过程被称为局部分配算法，如算法 6.4 所示。

算法 6.4 局部分配算法

局部变量： $\lambda, \lambda_{\text{new}}, a, a_{\text{new}}, V, V_{\text{new}}$

- 1: 根据最优策略 a_{new} 动态控制服务器速率;
 - 2: $a \leftarrow a_{\text{new}}$;
 - 3: $V \leftarrow V_{\text{new}}$;
 - 4: $\lambda \leftarrow \lambda_{\text{new}}$;
-

服务 Agent 作为代理和服务器之间的桥梁，维护服务的信息，并在二者之间传递优化结果和任务分发策略。代理 Agent 在所管理的服务范围内选择分配后，可得到最优的回报值的服务。顾客 Agent 一旦接收到服务请求，则激活所有与之联系的代理 Agent，请求各代理 Agent 尝试预分配该服务，根据代理 Agent 返回的回报值，选择最优的代理完成该服务。该决定完成后，代理 Agent 将任务实际分配到服务 Agent，进而分配给服务器 Agent，完成整个服务选择和速率控制过程。代理 Agent 上采用的全局最优查找算法如算法 6.5 所示；全局服务分配算法如算法 6.6 所示。

算法 6.5 全局优化算法

输入: I, J, P, λ_i for all $i \in I$

输出: ΔV_I

局部变量: $A, X_{m \times n} = [x_{ij}]_{m \times n}$

1: $x_{ij} \leftarrow 0$, for $\forall i \in I, \forall j \in J$;

2: $\Delta V_I \leftarrow 0$

3: 将 λ_i 倒序排序;

4: **for** $i \leftarrow 1$ **to** m **do**

5: $j_{\text{opt}} \leftarrow 0, \Delta V_{\text{opt}} \leftarrow -\infty$;

6: **for all** $j \in J_i$ **do**

7: 找到使得 $l_{jk} = 1$ 成立的参数 k ;

8: 在服务器 k 上基于输入参数 λ_i 执行局部优化算法, 得到输出 ΔV_j ;

9: **if** $\Delta V_j > \Delta V_{\text{opt}}$ **then**

10: $j_{\text{opt}} \leftarrow j, \Delta V_{\text{opt}} \leftarrow \Delta V_j$;

11: **end if**

12: **end for**

13: $x_{ij_{\text{opt}}} \leftarrow 1$;

14: $\Delta V_I \leftarrow \Delta V_I + \Delta V_{\text{opt}}$;

15: **end for**

16: **return** ΔV_I ;

算法 6.6 全局分配算法

局部变量: $A, X_{m \times n} = [x_{ij}]_{m \times n}$

1: **for** $i \leftarrow 1$ **to** m **do**

2: **for all** $j \in J_i$ **do**

3: **if** $x_{ij} = 1$ **then**

4: 找到使得 $l_{jk} = 1$ 成立的参数 k ;

5: 在服务器 k 上执行局部分配算法;

6: **end if**

7: **end for**

8: **end for**

6.5 案例分析

本节由真实的服务数据构建仿真实验, 对模型和算法进行验证分析。本案例选取了

QWS 数据^[117]。它对互联网上真实存在的 2507 个 Web 服务进行了测量,通过统计得到了响应时间、延迟、可用性等评价指标。这里选取其中的 1096 个服务,这些服务的功能存在重复(即有相同的服务提供同样的功能)。选取这些服务可便于讨论服务组合问题。根据这些服务的功能,将相同功能的服务归类,最终得到了 324 个类。对各类的服务数目进行统计,可以得到它的期望值为 3.38,每类服务数目的概率密度函数(Probability Density Function, PDF)曲线如图 6.4 所示。

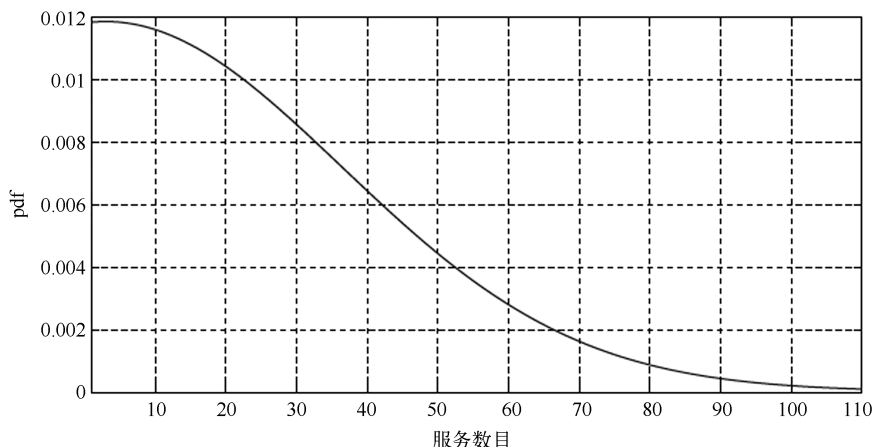


图 6.4 每类服务数目的 PDF 曲线

在模拟实验中,假设系统中共有 $p = 102$ 个服务器,在这些服务器上随机部署了 $n = 1096$ 个 Web 服务。用户需要完成一个包含 $m = 100$ 个服务的工作流。为了便于分析,突出算法的特性,假设这些服务器是同构的,它们的服务速率 $\bar{\mu}$ 由 QWS 数据集中的响应时间和延迟的统计数据计算得到。参照现今广泛使用的 CPU 速率控制技术^[103-106],CPU 可以动态调整主频以适应不同的负载需求,不仅可以降速工作,并且可以在一定范围内超频运行。因此,令服务器的服务速率控制器可以在离散集合 $\{0\%, 60\%, 100\%, 110\%\} \cdot \bar{\mu}$ 中自动选择合适的运行速率。由于 CPU 稳定运行的主频一般与电压成正比,因而式 (6.16) 中的参数 α 可以近似设置为 3^[26]。每个服务的平均收益设置为 $\bar{v} = 50$,电费设为 $\bar{c} = 10^{-4}$ 。每个决策周期的时间设为 $\tau = 10^{-3}$ s。服务的到达速率可以由数据集分析得到。

首先进行模拟实验,显示服务选择和速率控制优化得到的结果。优化算法执行完毕后,发现有 33 个服务器上在服务选择层面未被分配服务,即 $\lambda = 0$;且在速率控制层面被设置为始终处于睡眠状态,即 $\mu = 0$ 。这样,这些服务器可以长期处于不工作的状态,从而避免了长期运行所产生的能源开销。对于其他 69 个服务器,我们讨论其服务选择和速率控制结果。

图 6.5 所示为每个服务器上的动态速率控制策略。由图可见,最优的策略是在低负载(缓存队列中任务数较少)时运行在较低速的状态,以节省能耗开销;在较高的负载时,需要系统高速运行,以尽快完成任务,获取利润。该策略和当今的 CPU 动态主频控制策略的规律相一致。

图 6.6 所示为服务选择的结果。服务选择导致各服务器上的请求到达速率各不相同,并对速率控制策略造成一定的影响,因而各服务器上并非采用相同的速率控制,负载高低的变化阈值由服务选择的结果所决定。

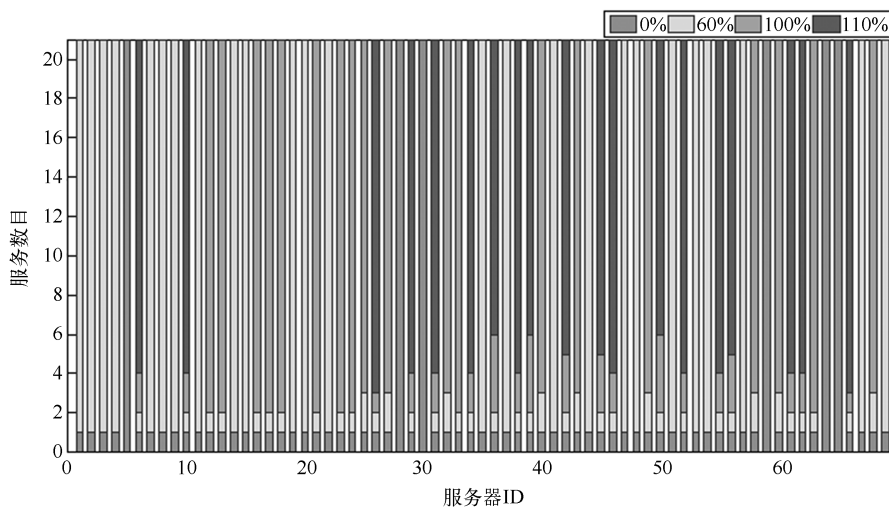


图 6.5 各服务器上的动态速率控制策略

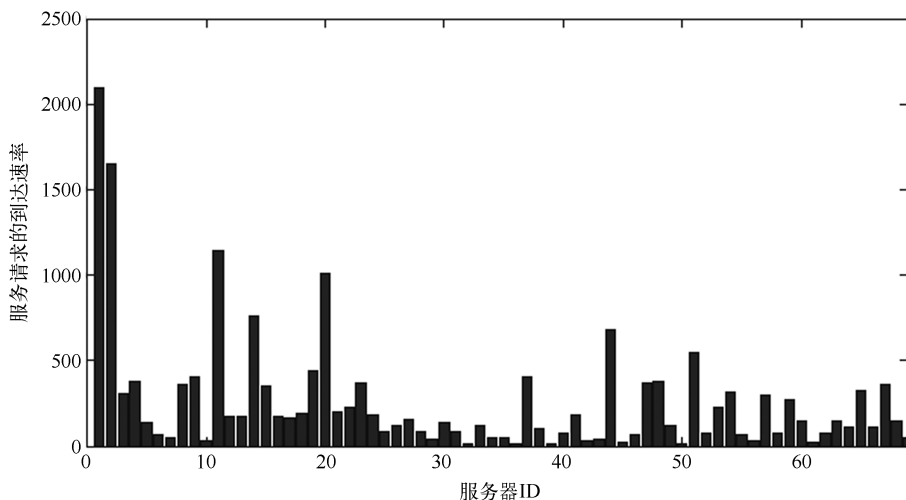


图 6.6 各服务器上的服务请求速率

随后，通过模拟实验，将本书所提出的优化算法与现有算法进行比较。考虑到目前尚未有研究同时考虑服务层和系统层进行能效优化，这里分别对仅考虑其中某一个方面的解决方案进行分析和比较。

首先对比仅考虑服务层的情况。此时服务速率被设置为定值 $100\% \cdot \bar{\mu}$ 。可以直接计算出每个服务器上的回报值，在服务选择时选取最优的回报。图 6.7 所示为每台服务器上的回报值比较情况。由图可见，数据点都在直线 $y = x$ 的上方，表示采用速率控制后可以进一步提高系统的能耗效率，提升服务得到的利润。

图 6.8 所示为每台服务器上的回报值差值 ΔR 的累积概率密度分布函数（Cumulative Distribution Function, CDF），揭示了在服务选择基础上增加速率控制后可以得到的进一步收益的情况。

然后对比仅考虑系统层未考虑服务组合的情况。在对比实验中，服务器上运行了基于

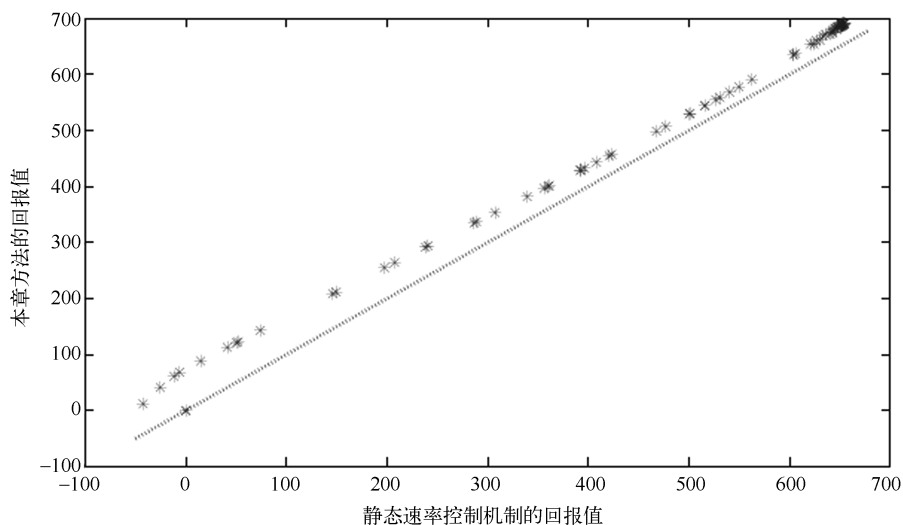


图 6.7 对比静态速率方案各服务器上的回报值

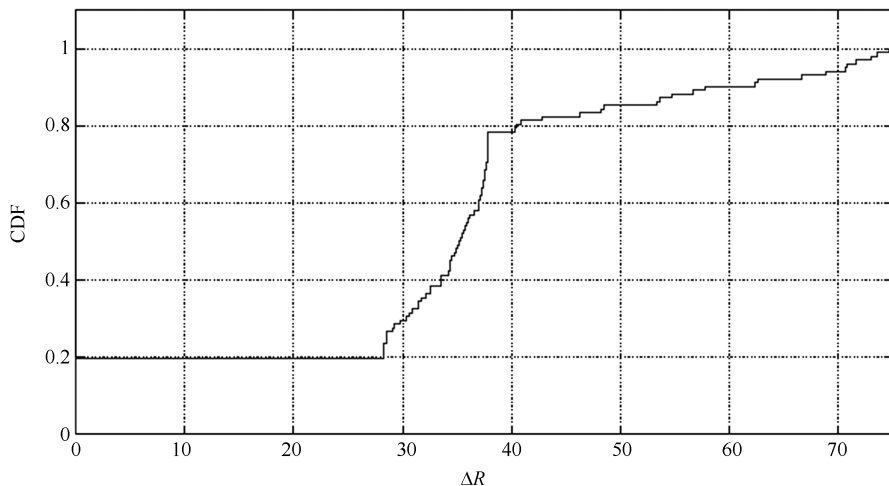


图 6.8 对比静态速率方案各服务器上回报值差值的 CDF 曲线

MDP 的动态服务速率控制算法，可以得到最佳的控制策略。为了便于直观分析，在服务层，对比实验采用了两个最常见的服务选择方法，即随机选择算法（Randomized Selection）和负载均衡算法（Load - Balanced Selection）。由于服务选择结果的差异，对单一服务器上的回报值的逐一对比讨论将不再有意义，因此这里讨论整个系统的总体回报值和所有服务器上回报值的分布情况。图 6.9 所示为本章提出的方法和其他已有算法的总体回报值的比较。可以看出，考虑服务层和系统层两个层面的解决方案，要比仅考虑任意其一的在能效上都更优。图 6.10 和图 6.11 所示为各服务器上的回报值分布情况。前者描绘了服务器上回报值的统计直方图，而后者显示了它们的 CDF 曲线。实验结果表明，本章所提出的优化算法中，高回报值的服务器在分布上更加集中，即更多的服务器处在较高收益的状态，因而可以得到更高的能源效率。综上所述，本章所提出的优化方法的有效性得到了验证。另一方面，实验结果显示，服务选择和速率控制对系统的能耗效率均有重要的作用，二者的综合考虑和优化要比仅从其中任一个方面进行优化得到更好的效果。

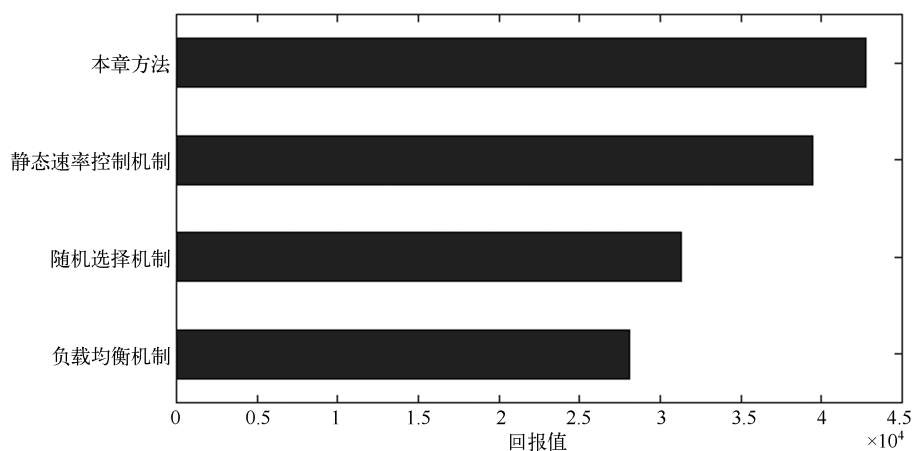


图 6.9 对比实验中的总体回报值比较

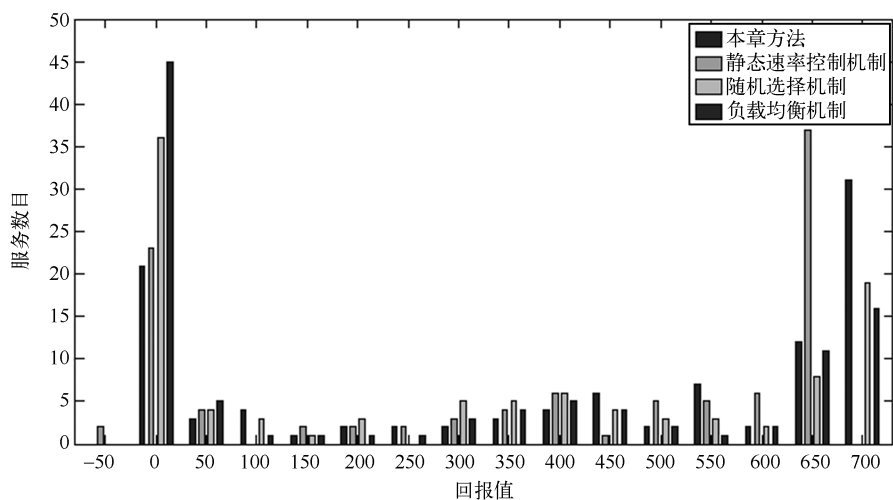


图 6.10 对比实验中的各服务器上的回报值分布情况

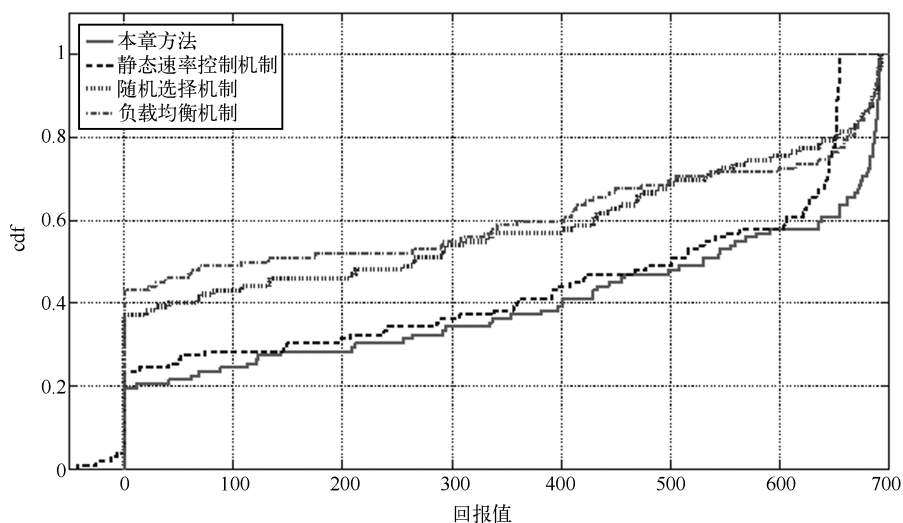


图 6.11 对比实验中的各服务器上回报值的 CDF 曲线

6.6 本章小结

本章对服务计算中的性能和能耗两个维度的属性进行了综合分析和优化。本章首先建立了服务计算系统的马尔可夫回报模型，阐述了模型的分析求解方法，给出了响应时间、利用率、收益、能耗开销的数学量化表达。同时，考虑服务层的服务选择和系统层的动态速率控制，建立了服务计算中的能效优化模型，基于马尔可夫决策过程给出了动态优化算法。理论上，证明了基于马尔可夫决策的优化方法得到结果的最优性，给出了马尔可夫决策模型和马尔可夫回报模型间对应关系的量化表达。进一步在大规模服务计算系统中，提出了基于多 Agent 技术的协同分布式优化框架及其优化算法，用于解决大规模服务中的服务选择和能耗控制的综合优化问题。最后，在实际的服务质量数据集上进行了模拟实验，验证了算法的有效性和最优性。

基于平均时间马尔可夫决策的 可信性 - 能耗综合优化

性能、可信性和能耗是服务计算中重要的 3 个评价维度，在服务计算中受到广泛的关注。如何在服务计算系统中保障服务的性能、可信性的同时，尽量减小能源消耗开销，是服务计算中亟待解决的研究问题。

在服务计算系统中，性能、可信性和能耗之间并非独立存在，它们之间具有复杂的相互影响和相互联系。单独对三者分别进行分析和优化，则可能忽略了三者之间的影响，造成求解过程的不准确，以及优化效果的不显著，因此需要在一个统一的模型中对这些指标综合分析和讨论，给出合适的优化路线，达到三者的折中优化。

目前，对服务计算的可信性讨论相对缺少，考虑可信性的能耗优化则更为少见。在其他类型的系统中，对可信性和能耗的相互关系和综合优化的研究尚处在起步阶段。例如，在嵌入式系统中，有研究指出，用于系统功耗控制的 DVFS 技术将影响系统的失效率，若单纯地降低系统能耗，将会造成系统不稳定，降低系统的可靠性^[118]。有学者在多处理器的实时系统中讨论了可靠性与能耗的关系，同样指出了降低能耗对系统可靠性的不利影响，进而提出了在保障可靠性的同时优化能耗效率的方法^[119]。在无线传感器网络中，有学者从网络和链路传输服务质量的角度讨论了可靠性与能耗之间的相互关系，并设计了二者折中的传输协议和机制^[120]。这些研究仅考虑了可信性中可靠性这一个属性，未从可信性的整个理论体系中进行研究，具有一定的片面性。另外，已有研究工作仅针对特定系统，如嵌入式系统、多处理器系统和传感器网络等，在服务计算领域的相关研究比较稀缺。对服务计算中可信性和能耗的研究，是目前该领域内的一个空白。

对此，本章内容旨在填补这一空白，即对服务计算系统中的可信性和能耗进行全面系统的研究，同时考虑服务过程中的性能收益，进行多属性的综合评价和优化。在已有的建模思路和方法的基础上，建立了服务计算系统的马尔可夫回报模型（MRM），该模型可以对系统的可信性和能耗同时进行评价。在模型基础上，基于平均时间马尔可夫决策过程对系统中的能效优化进行了建模，介绍了优化问题的求解方法，给出了该方法的最优性的理论证明。此外，为了应对服务大规模的特性，提出了 MRM 模型中的状态聚合方法，该方法可以有效减少状态空间，更直观地显示系统的特性。

7.1 服务系统的可依赖性—能耗评价模型

7.1.1 可信赖性模型

借鉴本书第4章中的建模思路，从面向服务的角度对系统进行建模，如图7.1所示。

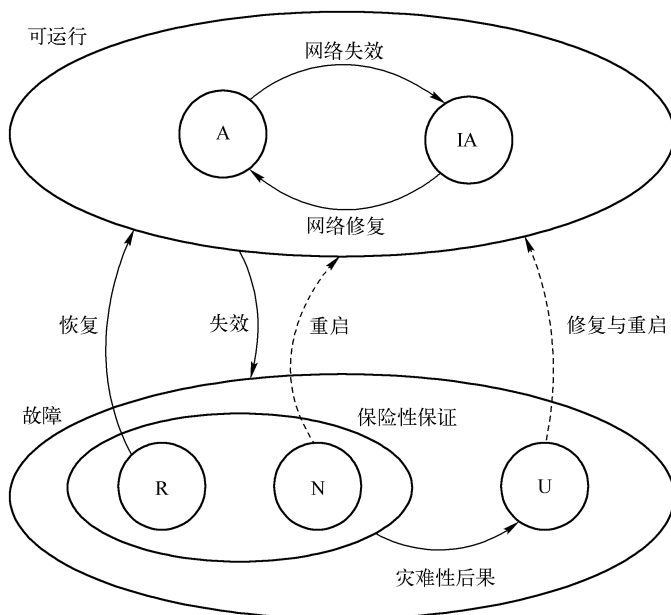


图 7.1 服务计算系统的简化 MRM 模型

这里，为了方便进一步的讨论，对第4章中的模型进行如下两点简化。

(1) 考虑到完整性也是安全性中的一个属性^[14,45]，而且实际系统设计中更关注和系统故障直接相关的可靠性、可用性、可维护性等属性，对完整性的关注程度不高^[59,85]。因此在本章中，略过了对完整性属性的讨论，这样可以更关注于系统部件失效对系统可信赖性和能耗的影响规律，且可以更准确地在实际系统日志数据^[85]上进行分析讨论。

(2) 本章使用连续时间马尔可夫回报模型对系统状态转移进行建模。虽然该模型的描述能力不如前面所述的半马尔可夫模型，在评价分析的精确性上有所牺牲，但在优化过程中却可以带来巨大的便利。在本章第7.3节将证明，该马尔可夫回报模型可以完全等价地转化为平均时间马尔可夫决策过程，从而可以用MDP算法高效地求解评价模型的优化问题，避免了求解马尔可夫链再进行优化所需要的巨大的计算开销。

在该MRM模型中，定义状态集合为 $S = \{A, IA, R, N, U\}$ ，将状态转移速率矩阵表示为

$$Q = \begin{bmatrix} q_{A,A} & q_{IA} & q_R & q_N & q_U \\ q_A & q_{IA,IA} & q_R & q_N & q_U \\ q_{RA} & q_{RIA} & q_{R,R} & 0 & q_{RU} \\ q_{NA} & q_{NIA} & 0 & q_{N,N} & q_{NU} \\ q_{UA} & q_{UIA} & 0 & 0 & q_{U,U} \end{bmatrix} \quad (7.1)$$

式中,

$$q_{i,i} = - \sum_{j \neq i} q_{i,j}, i \in S = \{A, IA, R, N, U\} \quad (7.2)$$

由马尔可夫模型理论可知, 可以通过以下方程组求解每个状态的稳态概率:

$$\begin{cases} \pi \cdot Q = 0 \\ \sum_{i \in S} \pi_i = 1 \end{cases} \quad (7.3)$$

参考第4章已经给出的可信赖性属性的定义, 即可对可信赖性进行量化评价。考虑到每个状态对应的属性语义, 参考前面 MTTF 分析中对状态进行分类的方法, 可以定义每个状态的可信赖性属性回报值, 见表 7.1。

表 7.1 可信赖性属性对应的 MRM 回报值定义

属 性	r_A	r_{IA}	r_R	r_N	r_U
可用性	1	0	0	0	0
可靠性	1	1	0	0	0
可维护性	1	1	1	0	0
保险性	1	1	1	1	0

7.1.2 能耗模型

图 7.1 所示的 MRM 不仅可以用于评价系统的可信赖性, 还可以对系统的功率和能耗进行评价。

在系统的运行过程中, 不同的状态可以对应不同的功耗状态。考虑式 (6.16) 中所示的功率, 如果将每个状态的回报值定义为该状态的功率, 即 $r_i = P_i$, 则系统整体的稳态功率 (即平均功率的期望值) 可以用稳态回报值表示, 如:

$$P = \sum_{i \in S} \pi_i \cdot r_i \quad (7.4)$$

那么, 系统在时间 t 内所消耗的能量期望值可以由下式计算得到:

$$E(t) = \int_0^t P(\tau) d\tau = P \cdot t = \sum_{i \in S} \pi_i \cdot r_i \cdot t \quad (7.5)$$

可以看出, 该 MRM 模型可以通过不同的回报值定义, 同时对系统的可信赖性和能耗两个维度的多个属性进行评价。作为一个整体, 可以在该模型上进行可信赖性和能耗的综合优化, 从而全面、系统地研究服务计算中能效优化的问题。

7.1.3 嵌入离散时间马尔可夫模型

同样, 由于计算机系统事件的离散特性, 这里将连续的 MRM 模型嵌入到一个离散时间马尔可夫链 (DTMC) 中。设嵌入的时间单位为 τ , 令该嵌入 DTMC 的转移概率矩阵为 $P_\tau = [p_{ij}]$, 则有:

$$p_{i,j} = \frac{q_{i,j}}{\sum_{k \neq i} q_{i,k}} \cdot (1 - e^{-\sum_{k \neq i} q_{i,k} \cdot \tau}) \quad i \neq j \quad (7.6)$$

$$p_{i,j} = e^{-\sum_{k \neq i} q_{i,k} \cdot \tau} \quad i = j \quad (7.7)$$

该嵌入链的稳态概率 $v = [v_A, v_{IA}, v_R, v_N, v_U]$ 由如下方程组计算:

$$\begin{cases} v = v \cdot P_T \\ \sum_{i \in S} v_i = 1 \end{cases} \quad (7.8)$$

同样可以证明, 嵌入 DTMC 和原始的 MRM 模型的稳态概率相等, 在回报值上, 两个模型也是等效的, 即

$$\pi_i = \frac{v_i h_i}{\sum_{j \in S} v_j h_j} = \frac{v_i}{\sum_{j \in S} v_j} = v_i \quad (7.9)$$

$$\bar{R}' = \sum_{i \in S} v_i \cdot r_i = \sum_{i \in S} \pi_i \cdot r_i = \bar{R} \quad (7.10)$$

7.2 基于回报值等价的状态聚合方法

在大规模的系统建模中, 状态数目过多、状态空间过大一直是影响模型分析效率的因素。将模型中的状态按照一定的标准进行聚合, 转化为状态数目较少的评价模型, 不仅利于模型的简化、分析、求解, 而且可以更直观地描述系统的动态行为, 揭示系统指标特性。本节将针对服务计算系统的离散时间马尔可夫回报模型, 从可信赖性模型和能耗模型两个角度出发, 分别研究模型的状态聚合方法。

7.2.1 可信赖性模型中的状态聚合

考虑到可信赖性的语义, 对于某一种属性, 状态可以被分为满足该属性和不满足该属性两类。例如, 对于可用性属性, 可访问 (A) 状态即为满足该属性的状态, 其他状态皆为不满足状态。而对于可靠性, 则可访问和不可访问都是满足可靠性的状态, 其他 3 个故障状态视为不满足可靠性的状态。可以根据该语义值定义每个状态的回报, 见表 7.1。在该 MRM 中, 回报值是离散的, 仅有 0 和 1 两种可能性。这样, 按照回报值的不同, 将状态分为开 (U) 和关 (D) 两类, 前者的回报值为 1, 后者为 0。类似文献 [121] 中的定义, 将 U 状态至 D 状态的变迁定义为红色变迁 (Red Transition), 表示故障失效情形; D 状态至 U 状态变迁为绿色变迁 (Green Transition), 代表故障修复过程。这样, 可以将复杂的多状态 MRM 模型聚合为 2 状态的简化模型, 如图 7.2 所示。

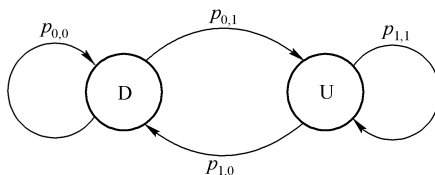


图 7.2 可信赖性的状态聚合模型

下面给出聚合模型中转移概率参数的计算方法。红色变迁的转移概率 $p_{1,0}$ 计算方法为

$$p_{1,0} = \frac{\sum_{i,j \in R} \pi_i \cdot p_{i,j}}{\pi_U} \quad (7.11)$$

式中,

$$\pi_U \equiv \sum_{k \in U} \pi_k \quad (7.12)$$

绿色变迁的转移速率的计算方法为

$$p_{0,1} = \frac{\sum_{i,j \in G} \pi_i \cdot p_{i,j}}{\pi_D} \quad (7.13)$$

式中,

$$\pi_D \equiv \sum_{k \in D} \pi_k \quad (7.14)$$

聚合模型中两个状态的自环变迁可以由马尔可夫链中每个状态的变迁概率之和为1的性质通过已有的计算结果得到,即

$$p_{0,0} = 1 - p_{0,1} \quad p_{1,1} = 1 - p_{1,0} \quad (7.15)$$

对于该状态聚合模型,两个状态的稳态概率可以由马尔可夫链稳态平衡方程得到,即

$$[\pi_0, \pi_1] = [\pi_0, \pi_1] \cdot \begin{bmatrix} 1 - p_{0,1} & p_{0,1} \\ p_{1,0} & 1 - p_{1,0} \end{bmatrix} \quad (7.16)$$

求解式(7.16),得到稳态概率的直观表达,即

$$\pi_0 = \frac{p_{1,0}}{p_{0,1} + p_{1,0}} \quad (7.17)$$

$$\pi_1 = \frac{p_{0,1}}{p_{0,1} + p_{1,0}} \quad (7.18)$$

进而,该聚合模型的稳态回报值的数学表达式为

$$\bar{R}_{eq} = \sum_i \pi_i \cdot r_i = \pi_1 \quad (7.19)$$

下面证明,聚合后的两状态模型和原模型具有完全相等的稳态回报值,即聚合方法在稳态可信赖性评价上是准确有效的。

【定理 7.1】 在稳态分析中,可信赖性的状态聚合模型和原模型的回报值是相等的,即

$$\bar{R}_{eq} = \bar{R} \quad (7.20)$$

证明: 首先,在稳态分析中,根据柯尔莫洛夫(Kolmogorov)方程,有

$$\pi_i = \sum_{j=1}^n \pi_j \cdot p_{j,i} \quad (7.21)$$

已知 $p_{i,i} = 1 - \sum_{j \neq i} p_{i,j}$, 结合式(7.21),可以得到:

$$\pi_i \cdot \sum_{j \neq i} p_{i,j} = \sum_{j \neq i} \pi_j \cdot p_{j,i} \quad (7.22)$$

不失一般性,假设满足属性的状态(U状态聚合前的各状态)编号为1到J,不满足的状态(D状态对应的状态)编号为(J+1)到n,推导得到:

$$\sum_{i=1}^J \pi_i \cdot \left(\sum_{j \neq i} p_{i,j} \right) = \sum_{i=1}^J \sum_{j \neq i} \pi_j \cdot p_{j,i} \quad (7.23)$$

$$\sum_{i=1}^J \pi_i \cdot \left(\sum_{k=1, k \neq i}^J p_{i,k} + \sum_{k=J+1}^n p_{i,k} \right) = \sum_{i=1}^J \left(\sum_{j=1, j \neq i}^J \pi_j \cdot p_{j,i} + \sum_{j=J+1}^n \pi_j \cdot p_{j,i} \right) \quad (7.24)$$

$$\sum_{i=1}^J \pi_i \cdot \sum_{k=1, k \neq i}^J p_{i,k} + \sum_{i=1}^J \pi_i \cdot \sum_{k=J+1}^n p_{i,k} = \sum_{i=1}^J \sum_{j=1, j \neq i}^J \pi_j \cdot p_{j,i} + \sum_{i=1}^J \sum_{j=J+1}^n \pi_j \cdot p_{j,i} \quad (7.25)$$

改变式 (7.25) 中求和变量的顺序, 可以发现该等式左右两侧的第一项相等, 因而有:

$$\sum_{i=1}^J \pi_i \cdot \sum_{k=J+1}^n p_{i,k} = \sum_{i=1}^J \sum_{j=J+1}^n \pi_j \cdot p_{j,i} \quad (7.26)$$

考虑到两类状态的编号顺序, 可以证明如下等式成立:

$$\sum_{i,j \in R} \pi_i \cdot p_{i,j} = \sum_{i,j \in G} \pi_i \cdot p_{i,j} \quad (7.27)$$

另外, 由式 (7.11) 和式 (7.13), 结合式 (7.17) 和式 (7.18), 有:

$$\frac{\pi_0}{\pi_1} = \frac{p_{1,0}}{p_{0,1}} = \frac{\sum_{i,j \in R} \pi_i \cdot p_{i,j}}{\sum_{i,j \in G} \pi_i \cdot p_{i,j}} \cdot \frac{\pi_D}{\pi_U} \quad (7.28)$$

进一步, 将式 (7.27) 和式 (7.28) 结合, 得到如下结论:

$$\frac{\pi_0}{\pi_1} = \frac{\pi_D}{\pi_U} \quad (7.29)$$

由于 $\pi_0 + \pi_1 = 1$ 和 $\pi_U + \pi_D = 1$, 进而可以证明式 (7.30) 和式 (7.31) 成立。从而定理得证。

$$\pi_0 = \pi_D \quad \pi_1 = \pi_U \quad (7.30)$$

$$\bar{R}_{eq} = \pi_1 = \pi_U = \sum_{k \in U} 1 \times \pi_k + \sum_{k \in D} 0 \times \pi_k = \bar{R} \quad (7.31)$$

7.2.2 能耗模型中的状态聚合

与可信赖性模型中的状态相比, 能耗模型的状态聚合过程更为复杂。系统在每个状态可以有不同的功耗值, 该值的取值范围是一个连续空间 (可以是实数集)。对于该类型的状态聚合, 采取类似的思路, 将状态聚合为高 (H) 和低 (L) 两个状态。H 至 L 状态的变迁定义为绿色变迁, 表示系统的功耗减小的过程。反之, L 至 H 定义为红色变迁, 代表功耗的提高。能耗的状态聚合模型如图 7.3 所示。

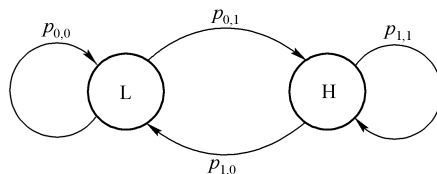


图 7.3 能耗的状态聚合模型

绿色变迁转移概率的计算方法为

$$p_{1,0} = \sum_{i,j \in G} \Pr\{s_i | H\} \cdot p_{i,j} \cdot (r_i - r_j) = \frac{\sum_{i,j \in G} \pi_i \cdot p_{i,j} \cdot (r_i - r_j)}{\pi_H} \quad (7.32)$$

式中,

$$\pi_H \equiv \bar{R} = \sum_{k \in S} \pi_k \cdot r_k \quad (7.33)$$

红色变迁转移概率的计算方法为

$$p_{0,1} = \sum_{i,j \in R} \Pr\{s_i | L\} \cdot p_{i,j} \cdot (r_j - r_i) = \frac{\sum_{i,j \in R} \pi_i \cdot p_{i,j} \cdot (r_j - r_i)}{\pi_L} \quad (7.34)$$

式中,

$$\pi_L \equiv 1 - \pi_H \quad (7.35)$$

通过求解马尔可夫链的稳态方程 $\pi = \pi P_T$, 得到该聚合模型的稳态概率表达, 即

$$\pi_0 = \frac{p_{1,0}}{p_{0,1} + p_{1,0}} \quad (7.36)$$

$$\pi_1 = \frac{p_{0,1}}{p_{0,1} + p_{1,0}} \quad (7.37)$$

同样, 定义 H 状态的回报值为 1, L 状态的回报值为 0, 则该聚合模型的稳态回报值为

$$\bar{R}_{eq} = \sum_i i \cdot \pi_i = 0 \cdot \pi_0 + 1 \cdot \pi_1 = \pi_1 \quad (7.38)$$

类似地, 通过定理 7.2, 证明该状态聚合方法的正确性。

【定理 7.2】 在稳态分析中, 能耗的状态聚合模型和原模型的回报值是相等的, 即

$$\bar{R}_{eq} \equiv \pi_1 = \pi_H \equiv \bar{R} \quad (7.39)$$

证明: 类似地, 可以通过证明 $\pi_0/\pi_1 \equiv p_{1,0}/p_{0,1} = \pi_L/\pi_H$ 成立来得到式 (7.39) 的结论。

由式 (7.32) 和式 (7.34) 可以得到, 如果式 (7.40) 成立, 则必有 $p_{1,0}/p_{0,1} = \pi_L/\pi_H$ 。

$$\sum_{i,j \in R} \pi_i \cdot p_{i,j} \cdot (r_j - r_i) = \sum_{i,j \in G} \pi_i \cdot p_{i,j} \cdot (r_i - r_j) \quad (7.40)$$

不失一般性, 将状态按照回报值从小到大的顺序重新编号, 即假设:

$$r_i \leq r_j, 1 \leq i \leq j \leq n \quad (7.41)$$

在式 (7.22) 等式两侧分别乘以 r_i , 可以得到:

$$\pi_i \left(\sum_{k \neq i} p_{i,k} \right) \cdot r_i = \sum_{j \neq i} \pi_j \cdot p_{j,i} \cdot r_i \quad (7.42)$$

令 $i=1, 2, \dots, n$, 将所有等式相加, 得到以下结论:

$$\sum_{i=1}^n \pi_i \cdot \left(\sum_{k \neq i} p_{i,k} \right) \cdot r_i = \sum_{i=1}^n \sum_{j \neq i} \pi_j \cdot p_{j,i} \cdot r_i \quad (7.43)$$

将上式进行变形, 进而推导得到:

$$\sum_{i=1}^n \pi_i \cdot \left(\sum_{k=1}^{i-1} p_{i,k} + \sum_{k=i+1}^n p_{i,k} \right) \cdot r_i = \sum_{i=1}^n \left(\sum_{j=1}^{i-1} \pi_j \cdot p_{j,i} + \sum_{j=i+1}^n \pi_j \cdot p_{j,i} \right) \cdot r_i \quad (7.44)$$

$$\sum_{i=2}^n \pi_i \cdot \sum_{k=1}^{i-1} p_{i,k} r_i + \sum_{i=1}^{n-1} \pi_i \cdot \sum_{k=i+1}^n p_{i,k} r_i = \sum_{i=2}^n \sum_{j=1}^{i-1} \pi_j p_{j,i} r_i + \sum_{i=1}^{n-1} \sum_{j=i+1}^n \pi_j p_{j,i} r_i \quad (7.45)$$

$$\sum_{i=1}^{n-1} \pi_i \cdot \sum_{k=i+1}^n p_{i,k} r_i - \sum_{i=2}^n \sum_{j=1}^{i-1} \pi_j p_{j,i} r_i = \sum_{i=1}^{n-1} \sum_{j=i+1}^n \pi_j p_{j,i} r_i - \sum_{i=2}^n \pi_i \cdot \sum_{k=1}^{i-1} p_{i,k} r_i \quad (7.46)$$

改变式 (7.46) 等号两侧第二项中的求和顺序, 可以得到:

$$\sum_{i=1}^{n-1} \pi_i \sum_{k=i+1}^n p_{i,k} r_i - \sum_{j=1}^{n-1} \pi_j \sum_{i=j+1}^n p_{j,i} r_i = \sum_{i=1}^{n-1} \sum_{j=i+1}^n \pi_j p_{j,i} r_i - \sum_{k=1}^{n-1} \sum_{i=k+1}^n \pi_i p_{i,k} r_i \quad (7.47)$$

进而有:

$$\sum_{i=1}^{n-1} \pi_i \sum_{k=i+1}^n p_{i,k} \cdot (r_i - r_k) = \sum_{i=1}^{n-1} \sum_{j=i+1}^n \pi_j \cdot p_{j,i} \cdot (r_i - r_j) \quad (7.48)$$

因此可以证明式 (7.40) 成立, 从而定理 7.2 得证。

7.3 基于马尔可夫决策的可信赖功耗控制算法

当今计算机系统提供的 DVFS 动态功耗控制机制为系统能耗效率优化提供了技术支持。如何根据系统状态动态调整系统的电压和速率, 成为了计算机系统能耗优化中的重要科学问题。由于电压控制往往在硬件层实现, 对用户的开放程度较低, 且电压和系统运行主频呈正相关的关系^[26], 因此本节仅讨论对系统速率进行优化控制的问题。

7.3.1 优化建模与算法

在服务计算中, 考虑到系统的状态转移情况, 将动态能耗控制形式化为一个马尔可夫决策过程 (MDP)。该 MDP 的决策周期、状态空间、决策空间、回报函数和优化目标的定义如下所述。

1) 决策周期 每个决策周期的长度为 τ , 按照时间顺序编号为 $n \in \{1, 2, \dots\}$ 。对应地, 每次决策发生在时间 $t = \tau, 2\tau, \dots$ 。

2) 状态空间 在第 n 个决策周期的状态 $S(n)$ 定义为当前时间服务计算系统所处的状态, 即 $S(n) \in S = \{A, IA, R, N, U\}$ 。

3) 决策空间 第 n 个决策周期, 控制器需要从 k 个离散值中选取最优的服务速率。决策行为表示为 $a_n \in A = \{\mu_1, \mu_2, \dots, \mu_k\}$ 。

4) 回报函数 综合考虑到系统的性能、可信赖性和能耗, 将马尔可夫模型中每个状态的回报值定义为服务获取的利润, 即收益减去开销。收益由完成服务获得, 而开销主要由能源消耗产生。由系统状态的定义可知, 在可运行状态 (即可访问 A 状态和不可访问 IA 状态), 系统可以处理已接受的服务, 从而获得服务收益。设每个服务可以得到的收益值为 v_0 , 那么对于每一个决策周期, 在可运行状态服务的收益值为 $v_i = v_0 \mu_i \tau$ 。而在故障状态, 系统则无法提供任何服务, 因而该状态下仅有开销而没有收益, 回报值为负值, 系统运行的能耗开销可以表达为 $c_i = c_0 P_{\mu_i} \tau$, 其中 c_0 为单位能源消耗的价格。综上所述, 回报值可以形式化定义为

$$r(s, \mu_i) = \begin{cases} v_0 \mu_i \tau - c_0 P_{\mu_i} \tau, & s \in \{A, IA\} \\ -c_0 P_{\mu_i} \tau, & s \in \{R, N, U\} \end{cases} \quad (7.49)$$

5) 优化目标 与经济学概念不同, 可信赖性的讨论一般不会区别定义现在和未来的回报值。因为无论何时发生系统故障, 对系统的功能都是严重的影响; 尤其是一些关键部件的故障, 产生的后果是灾难性的。所以, 与第 6 章中的优化目标不同, 本章考虑无折扣因子的平均时间回报值优化问题, 形式化表示为稳态回报期望值最大化, 即

$$\underset{a \in A}{\text{maximize}} \quad \lim_{N \rightarrow \infty} \frac{1}{N} E \left[\sum_{k=1}^N r(S(k), a) \right] \quad (7.50)$$

无折扣因子的平均时间马尔可夫决策过程是比较特殊的一类 MDP, 它可由相关值迭代算法 (Relative Value Iteration)^[112]得到每个状态的最优能耗控制策略。该算法的具体流程如算法 7.1 所示。

算法 7.1 相关值迭代算法

```

1: 选取  $V_0, s^* \in S, \varepsilon > 0$ ;
2:  $W_0 \leftarrow V_0 - V_0(s^*) \vec{e}$ ;
3: for  $n \leftarrow 0$  to  $\infty$  do
4:   for all  $s \in S$  do
5:      $V_{n+1}(s) \leftarrow \max_{a \in A} \{ r(s, a) + \sum_{s' \in S} p(s' | s, a) W_n(s') \}$ ;
6:   end for
7:    $W_{n+1} \leftarrow V_{n+1} - V_{n+1}(s^*) \vec{e}$ ;
8:   if  $\max_{s \in S} [ | V_{n+1}(s) - V_n(s) | ] < \varepsilon$  then
9:      $a_\varepsilon(s) \leftarrow \operatorname{argmax}_{a \in A} \{ r(s, a) + \sum_{s' \in S} p(s' | s, a) V_{n+1}(s') \}$ ;
10:    return  $a_\varepsilon$ ;
11:   end if
12: end for

```

7.3.2 最优性证明

首先证明 MDP 优化建模对回报模型优化的有效性。

【定理 7.3】当最优策略 $a = a^*$ 找到后，MDP 优化目标中的回报值和马尔可夫回报模型中的回报值完全等价，即

$$\lim_{N \rightarrow \infty} \frac{1}{N} \mathbf{E} \left[\sum_{k=1}^N r(S(k)) \right] = \sum_{j \in S} r_j \pi_j \quad (7.51)$$

证明：令 $M_j(N)$ 表示 MRM 模型的离散时间马尔可夫链在时间区间 $1, 2, \dots, N$ 中对状态 j 的访问次数，因而有：

$$\sum_{k=1}^N r(S(k)) = \sum_{j \in S} M_j(N) r(j) \quad (7.52)$$

由大数定律（Law of Large Numbers, LLN）可得 MRM 模型稳态概率的表达形式为

$$\lim_{N \rightarrow \infty} \frac{M_j(N)}{N} = \pi_j \quad (7.53)$$

对式 (7.52) 左右两侧分别除以 N ，取期望，并令 $N \rightarrow \infty$ ，有

$$\lim_{N \rightarrow \infty} \mathbf{E} \left[\frac{1}{N} \sum_{k=1}^N r(S(k)) \right] = \lim_{N \rightarrow \infty} \frac{1}{N} \mathbf{E} \left[\sum_{k=1}^N r(S(k)) \right] \quad (7.54)$$

$$\lim_{N \rightarrow \infty} \mathbf{E} \left[\frac{1}{N} \sum_{j \in S} M_j(N) r(j) \right] = \mathbf{E} [\pi_j r(j)] = \sum_{j \in S} r_j \pi_j \quad (7.55)$$

综上所述，可以证明式 (7.51) 成立，定理 7.3 得证。

下面证明算法 7.1 可以得到最优的回报值。

【定理 7.4】算法 7.1 可以得到该 MDP 的 ε -最优解。

证明：文献 [122] 讨论了有限状态空间和决策空间的平均时间 MDP，并提出了平均开销优化方程（Average Cost Optimality Equation, ACOE），见式（7.56）。式中 ρ 为一个常量。该文献中证明，如果求解算法能够达到等式右侧的最大值，即可保证 MDP 在稳态的最优性。

$$\rho + h(s) = \max_{a \in A} \left\{ r(s, a) + \sum_{s' \in S} p(s' | s, a) h(s') \right\} \quad (7.56)$$

对于算法 7.1，令 $\rho = V(s^*)$ ， $h(s) = W(s)$ ，则有 $V(s) = V(s^*) + W(s) = \rho + h(s)$ 。从而证明该算法满足 ACOE 方程，定理 7.4 得证。

7.4 案例分析

为了直观显示模型评价和优化策略，本节在服务计算系统的日志数据集上进行数值模拟实验。这里，仍然使用美国洛斯阿拉莫斯国家实验室（Los Alamos National Laboratory, LANL）提供的大规模系统故障日志^[85]。由系统故障日志得到状态变迁的时间序列，通过分析其中的时间数据，统计得到马尔可夫模型参数，包括失效速率、修复速率等。能耗参数选用 6.5 节所使用的参数值，以使本书各章节的实验设置相互统一。

首先分析系统的可信赖性。4.5 节已对该系统的可信赖性属性进行了分析，这里给出模型状态聚合后的结果。应用状态聚合的计算方法，根据表 7.1 中不同可信赖性属性的回报值定义，得到聚合状态模型中红色变迁和绿色变迁的转移概率，分别如图 7.4 和图 7.5 所示。由图可见，不同系统的模型参数呈现较大差异，这一点与第 4 章中所分析得到的结论相一致。

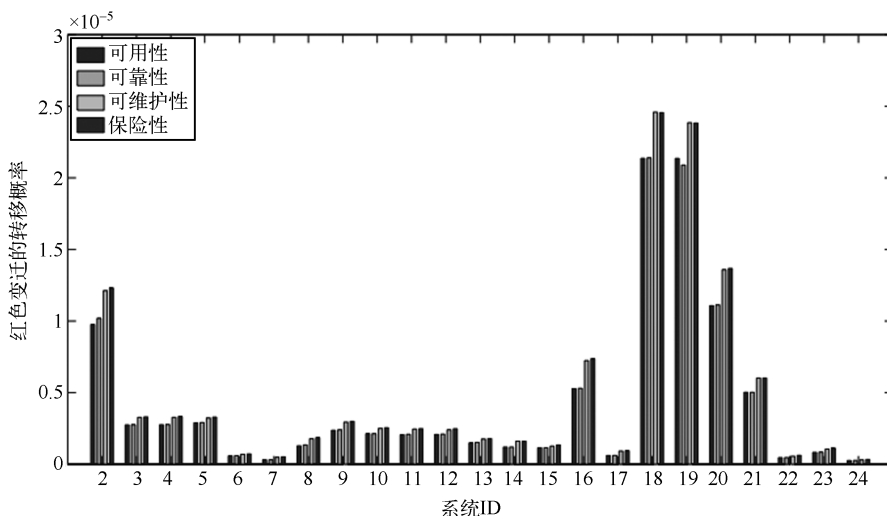


图 7.4 各属性红色变迁的转移概率

随后对该模型进行 MDP 的优化求解，通过算法 7.1 得到每个状态的最佳运行速率，如图 7.6 所示。

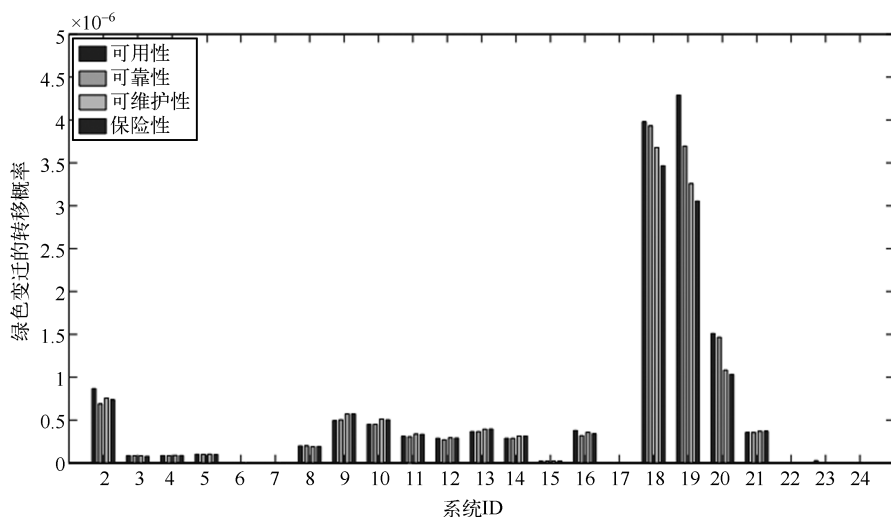


图 7.5 各属性绿色变迁的转移概率

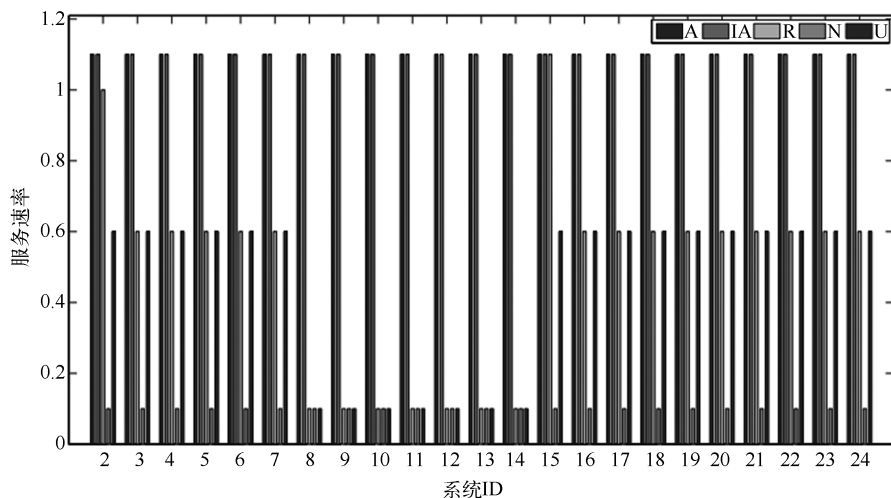


图 7.6 最优速率控制策略

综合考虑系统的性能收益、可信赖性影响和能耗开销后，为了达到最大收益，系统在不同的状态应运行在不同的速率上。各状态间不同的转移概率和性能收益决定了最优速率的选取。例如，系统在可运行状态可以持续完成服务，因而最优速率大都为运行在100%~110%的高速率状态，以获得更多的服务收益；而在故障状态，由于无法获得收益，系统则可能运行在低速状态。尤其是在不可修复的故障状态，由于系统外界的基础设施发生了问题，系统无法自动修复；从数据中可以看出，该类型故障修复时间很长，因而最优的策略是让系统处于低速状态，以减少能量的浪费。

7.5 本章小结

随着服务计算的不断发展，除性能外，可信赖性和能耗受到越来越多的关注。本章综合

考虑了性能收益、可信赖性和能耗，讨论了服务计算系统的建模理论和优化方法。提出了基于马尔可夫回报理论的随机模型，利用不同回报值的定义，该模型可以同时可对可信赖性和能耗两个维度的多个属性进行评价。在模型分析上，提出了离散时间马尔可夫回报模型的状态聚合方法，并给出了理论证明。结合马尔可夫回报模型，将系统中的动态速率控制形式化为一个平均时间马尔可夫决策问题，给出了该问题的求解算法。在理论上，证明了马尔可夫回报模型和马尔可夫决策模型在回报值上的等价性，给出了相关的定理。最后，结合系统故障数据进行了模拟实验的案例分析。

总结与展望

8.1 总 结

服务计算凭借其灵活、高效、易部署、易管理等方面的优势,逐渐成为 IT 基础设施的发展趋势。随着基于互联网的应用需求迅速提升,服务计算被广泛应用于各个领域,展现出极大的发展潜力。与此同时,伴随着服务数目的增长,为了对不同的服务和系统进行比较,对新系统进行有效设计,对已有系统进行改进,服务计算的评价和优化成为了学术界和工业界共同关注的热点问题。考虑到目前服务 QoS 的多方面需求,本书从性能、可信赖性和能耗 3 个维度出发,对服务计算中的多属性评价模型、分析方法和优化技术展开了深入研究。本研究工作的主要贡献和创新性如下所述。

1) 研究了服务计算中性能指标的建模和评价方法 从面向服务的角度出发,建立了原子服务的排队模型和服务系统的排队网络模型。针对不同的任务请求到达分布,给出了排队模型的量化分析方法。以模型分析结果为基础,提出了服务质量优化的建模方法,并以资源管理和任务调度为例,介绍了优化问题的求解方法。最后,以真实数据构建模拟实验,验证了模型的有效性和理论分析的正确性。该工作较为系统地研究了服务计算的性能模型,提出了建模方法、求解思路和基本的优化方法,为服务质量性能维度指标的评价和优化提供了理论参考。

2) 全面、系统地研究了服务计算系统的可信赖性 从可信赖性的指标体系出发,建立了服务计算系统的半马尔可夫评价模型。研究了模型的求解方法,提出了模型分析和化简技术,从稳态概率和时间分析两个维度,给出了可信赖性各属性的概率表达和平均失效时间分析。深入研究了部分可信赖性属性之间的相互关系,给出了这些关系的量化表达,以及相关的理论证明。进一步,将模型应用于服务选择和服务组合,提出了基于框图模型的可信赖性建模分析方法。最后,在大规模的系统故障日志上进行了实例讨论,并给出了敏感度分析方法,揭示了发现系统瓶颈的路线。该工作系统地研究了服务计算中的可信赖性指标体系,提出了服务计算系统可信赖性的建模评价方法和分析求解技术,为可信赖的服务计算系统的评价和优化提供了理论基础。

3) 研究了大规模服务排序评价技术 为了适应服务计算中服务数目规模巨大的特点,针对服务选择、服务推荐等需要解决的服务排序问题,提出了协同分布式的服务排序评价方

法；针对服务比较过程，建立了概率模型，并将服务排序形式化为一个有向图上的随机游走问题，利用马尔可夫模型理论对该问题进行求解。在实际应用中，以理论模型为基础，提出了将部分排序聚合为全局排序的算法，并给出了算法性能的理论分析和实验验证。可以证明，该服务评价技术可以显著减轻每个测试端的评价负担，大幅降低服务排序过程中的评价成本，并同时能够保证排序结果的准确性。该工作对于大规模的服务评价具有一定的理论意义和实用价值。

4) 研究了服务计算系统中性能和能耗的相互关系及综合优化问题 为了优化服务过程中的能耗效率，从服务层和系统层两个角度同时出发，提出了服务选择和动态速率控制综合优化方法。具体地，建立了服务计算系统的马尔可夫回报模型，对服务的性能和能耗进行量化分析。利用马尔可夫决策过程对动态速率控制进行建模，给出了优化方法，并从理论上证明了马尔可夫决策过程对马尔可夫回报模型优化的有效性。在大规模服务系统中，提出了基于多 Agent 技术的协同分布式优化框架，用以高效地对服务选择和动态速率控制同时进行优化。理论分析和实验结果均验证了算法的有效性。在优化思路，同时考虑服务层和系统层，进行综合优化，得到了比已有机制更优的能耗效率。在优化技术上，利用多 Agent 技术和马尔可夫决策优化方法，保证了求解过程的效率。

5) 研究了服务计算系统的性能、可信赖性和能耗的综合优化问题 建立了服务计算系统的马尔可夫回报模型，利用不同属性对各状态的回报值定义不同，在模型中实现了同时对多个属性进行评价的功能。为了简化模型分析和求解过程，提出了离散时间马尔可夫回报模型的状态聚合方法，针对离散型回报值和连续性回报值两种情况，分别给出了理论分析和证明。考虑服务收益和可信赖性的需求，将可信赖的能耗控制形式化为一个平均时间马尔可夫决策过程，给出了优化方法，并在理论上证明了该方法的有效性。这一工作是对马尔可夫回报模型理论的发展，在建模思路、分析技术和优化理论上具有可借鉴意义。

8.2 研究展望

服务计算作为一个新兴的领域，受到了学术界和工业界共同的关注。在以往的研究中，常常关注于服务计算中标准的制定、服务的部署、系统的设计等问题。而随着服务数目的增多，服务计算的广泛应用，服务和服务计算系统的评价和优化，成为了服务计算领域中一个重要的研究问题。本书围绕服务计算的多属性评价和综合优化问题进行了探索性工作，取得了一些创新性成果。但是，由于该问题的研究尚处于起步阶段，在本书的基础上继续开展更进一步的研究和深入探索仍具有价值和意义。在未来的研究中，可以从以下 4 点进一步开展工作。

(1) 在指标维度上，本书主要对性能、可信赖性和能耗 3 个维度的多个属性进行了讨论。然而，安全性也是某些应用服务所关注和需求的重要属性。在进一步的研究中，可以从安全性维度进行讨论，研究针对机密性、可认证性等属性的建模分析和优化方法。由于安全性的讨论中需要全面考虑用户（或攻击者）与系统之间的交互过程，因此本书中的面向服务、面向用户的建模方法将具有一定的借鉴价值。

(2) 在建模理论上，本书的建模方法主要基于马尔可夫或半马尔可夫模型。然而在某些更复杂的情况下，系统的动态行为可能不再具有马尔可夫特性或马尔可夫嵌入点特性，如在

分析底层网络流量过程中可能会遇到负载呈重尾分布或自相似分布等情况，造成模型分析求解的难度。因此在进一步的研究中，可以研究随机模型中非马尔可夫过程求解方法，利用随机分布近似等理论，给出准确的或近似的模型分析方法。

(3) 在优化思路，本书主要从马尔可夫回报模型出发，研究基于马尔可夫决策的优化方法。在进一步的研究中，可以考虑应用如李雅普诺夫优化 (Lyapunov Optimization) 等其他的随机优化方法，并在理论上研究随机模型和优化方法的对应关系。

(4) 在实验验证上，本书主要基于大规模实际系统或服务的数据进行仿真验证分析。在进一步的研究中，可以构建真实的服务计算系统作为实验平台，进而可以将本书所提出的模型和算法在实际系统中实现，验证模型的准确性和算法的有效性。此外，系统运行过程中所监控获取的真实数据可以进一步对系统和用户行为建模、优化方法的设计和改进具有一定的指导意义。

参考文献

- [1] I. Foster. Service – oriented science. *Science*, 2005, vol. 308, no. 5723: 814 – 817
- [2] L. J. Zhang, J. Zhang, and H. Cai. *Services Computing*. Beijing: Springer and Tsinghua university Press, 2007
- [3] L. J. Zhang. Services computing: A new discipline. *International Journal of Web Services Research (JWSR)*, 2005, vol. 2, no. 1: Editorial Preface
- [4] L. J. Zhang. EIC editorial: Introduction to the knowledge areas of services computing. *IEEE Transactions on Services Computing*, vol. 1, no. 2: 62 – 74, First 2008
- [5] C. Guo, H. Wu, K. Tan, L. Shi, Y. Zhang, and S. Lu. Dcell: A scalable and fault – tolerant network structure for data centers. in *Proceedings of the ACM SIGCOMM 2008 Conference on Data Communication*, ser. SIGCOMM’08, 2008: 75 – 86
- [6] R. Buyya, C. S. Yeo, and S. Venugopal. Market – oriented cloud computing: Vision, hype, and reality for delivering it services as computing utilities. in *Proceedings of the 10th IEEE International Conference on High Performance Computing and Communications (HPCC 2008)*, Sep. 2008: 5 – 13
- [7] S. Majithia, D. Walker, and W. Gray. A framework for automated service composition in service – oriented architectures. in *The Semantic Web: Research and Applications*, ser. *Lecture Notes in Computer Science*, C. Bussler, J. Davies, D. Fensel, and R. Studer, Eds. Springer Berlin Heidelberg, 2004, vol. 3053: 269 – 283
- [8] L. J. Zhang and M. Jeckle. The next big thing: Web services collaboration. in *Web Services – ICWS – Europe 2003*, ser. *Lecture Notes in Computer Science*, M. Jeckle and L. J. Zhang, Eds. Springer Berlin Heidelberg, 2003, vol. 2853: 1 – 10
- [9] E. Christensen, F. Curbera, G. Meredith, and S. Weerawarana. *Web Services Description Language (WSDL) 1.1*. [Online]. Available: <http://www.w3.org/TR/wsdl>
- [10] P. Ruth, X. Jiang, D. Xu, and S. Goasguen. Virtual distributed environments in a shared infrastructure. *Computer*, vol. 38, no. 5: 63 – 69, May 2005
- [11] C. Clark, K. Fraser, S. Hand, J. G. Hansen, E. Jul, C. Limpach, I. Pratt, and A. Warfield. Live migration of virtual machines. in *Proceedings of the 2nd Conference on Symposium on Networked Systems Design (NSDI 2005)*, ser. NSDI’05, 2005: 273 – 286
- [12] M. R. Marty and M. D. Hill. Virtual hierarchies to support server consolidation. in *Proceedings of the 34th Annual International Symposium on Computer Architecture (ISCA 2007)*, ser. ISCA’07, 2007: 46 – 56
- [13] L. Zeng, B. Benatallah, A. H. H. Ngu, M. Dumas, J. Kalagnanam, and H. Chang. QoS – aware middleware for web services composition. *IEEE Transactions on Software Engineering*, vol. 30, no. 5: 311 – 327, 2004
- [14] A. Avizienis, J. C. Laprie, B. Randell, and C. Landwehr. Basic concepts and taxonomy of dependable and secure computing. *IEEE Transactions on Dependable and Secure Computing*, vol. 1, no. 1: 11 – 33, Jan. – Mar. 2004

- [15] P. Bartalos and M. Blake. Green web services: Modeling and estimating power consumption of web services. in Proceedings of IEEE 19th International Conference on Web Services (ICWS 2012), Jun. 2012: 178 – 185
- [16] A. Keller and H. Ludwig. The wsla framework: Specifying and monitoring service level agreements for web services. Journal of Network and Systems Management, vol. 11, no. 1: 57 – 81, Mar. 2003
- [17] R. K. Jain. The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling. New York: Wiley, 1991
- [18] 林闯, 李雅娟, 王忠民. 性能评价形式化方法的现状和发展. 电子学报, Vol. 30, no. 12A: 1917 – 1922, Dec. 2002
- [19] 林闯. 计算机网络和计算机系统的性能评价. 北京: 清华大学出版社, 2001
- [20] W. Nie, J. Zhang, and K. J. Lin. Estimating real – time service process response time using server utilizations. in Proceedings of 2010 IEEE International Conference on Service – Oriented Computing and Applications (SOCA 2010), Dec. 2010: 1 – 8
- [21] Z. Zhou, F. Liu, H. Jin, B. Li, B. Li, and H. Jiang. On arbitrating the power – performance tradeoff in saas clouds. in Proceedings of IEEE INFOCOM 2013, Apr. 2013: 872 – 880
- [22] S. He, L. Guo, M. Ghanem, and Y. Guo. Improving resource utilisation in the cloud environment using multivariate probabilistic models. in Proceedings of IEEE 5th International Conference on Cloud Computing (CLOUD 2012), Jun. 2012: 574 – 581
- [23] J. C. Laprie, A. Avizienis, and H. Kopetz. Eds., Dependability: Basic Concepts and Terminology. Secaucus, NJ, USA: Springer – Verlag New York, Inc., 1992
- [24] B. Helvik. Perspectives on the dependability of networks and services. Teletronikk, vol. 100: 27 – 44, 2004
- [25] J. G. Koomey. Estimating total power consumption by servers in the US and the world. Lawrence Berkley National Laboratory, Tech. Rep., Feb. 2007
- [26] A. Beloglazov, R. Buyya, Y. Lee, and A. Zomaya. A taxonomy and survey of energy – efficient data centers and cloud computing systems. Advances in Computers, 2011, vol. 82, no. 2: 47 – 111
- [27] R. Brown, E. Masanet, B. Nordman, B. Tschudi, A. Shehabi, J. Stanley, J. Koomey, D. Sartor, P. Chan, J. Loper et al. Report to congress on server and data center energy efficiency: Public law 109 – 431. Lawrence Berkeley National Laboratory, Tech. Rep., Aug. 2007
- [28] Google Green. [Online]. Available: <http://www.google.com/green/>
- [29] J. Koomey. Growth in data center electricity use 2005 to 2010. Analytics Press, Aug. 2011
- [30] E. Feller, C. Rohr, D. Margery, and C. Morin. Energy management inIaaS clouds: A holistic approach. in Proceedings of IEEE 5th International Conference on Cloud Computing (CLOUD 2012), Jun. 2012: 204 – 212
- [31] S. Kikuchi and Y. Matsumoto. Performance modeling of concurrent live migration operations in cloud computing systems using prism probabilistic model checker. in Proceedings of IEEE International Conference on Cloud Computing (CLOUD 2011), July 2011: 49 – 56
- [32] C. Dabrowski and F. Hunt. Using markov chain analysis to study dynamic behaviour in large – scale grid systems. in Proceedings of the 7th Australasian Symposium on Grid Computing and e – Research (Aus – Grid 2009), ser. AusGrid '09. Darlinghurst, Australia, Australia: Australian Computer Society, Inc., 2009: 29 – 40
- [33] T. Thein, S. Chi, and J. Park. Availability modeling and analysis on virtualized clustering with rejuvenation. International Journal of Computer Science and Network Security, vol. 8, no. 9: 72 – 80, 2008
- [34] B. Guenter, N. Jain, and C. Williams. Managing cost, performance, and reliability tradeoffs for energy – a

- ware server provisioning. in Proceedings of IEEE INFOCOM 2011, Apr. 2011: 1332 – 1340
- [35] G. Bolch, S. Greiner, H. de Meer, and K. S. Trivedi. Queueing Networks and Markov Chains: Modeling and Performance Evaluation with Computer Science Applications. Hoboken, New Jersey, USA: John Wiley & Sons, 2006
- [36] R. Mullubhatla and K. Pattipati. Discrete – time markov reward models of automated manufacturing systems with multiple part types and random rewards. IEEE Transactions on Robotics and Automation, Oct. 2000, vol. 16, no. 5: 553 – 566
- [37] C. K. Chan. Availability analysis of load – sharing systems. in Proceedings of Annual Reliability and Maintainability Symposium (RAMS 2003), 2003: 551 – 555
- [38] N. Sato and K. Trivedi. Accurate and efficient stochastic reliability analysis of composite services using their compact markov reward model representations. in Proceedings of 2007 IEEE International Conference on Services Computing (SCC 2007), July 2007: 114 – 121
- [39] B. Madan, K. Gogeva – Popstojanova, K. Vaidyanathan, and K. Trivedi, “Modeling and quantification of security attributes of software systems. in Proceedings of International Conference on Dependable Systems and Networks (DSN 2002), 2002: 505 – 514
- [40] Q. Nguyen and A. Sood. Improving resilience of soa services along space – time dimensions. in Proceedings of IEEE/IFIP 42nd International Conference on Dependable Systems and Networks Workshops (DSN – W 2012), Jun. 2012: 1 – 6
- [41] 黄晓璐, 闵应骅, 吴起. 网络流量的半马尔可夫模型. 计算机学报, Oct. 2005, vol. 28, no. 10: 1592 – 1600
- [42] R. Calheiros, R. Ranjan, and R. Buyya. Virtual machine provisioning based on analytical performance and qos in cloud computing environments. in Proceedings of International Conference on Parallel Processing (ICPP 2011), Sep. 2011: 295 – 304
- [43] R. Birke, L. Chen, M. Gribaudo, and P. Piazzolla. Characterization analysis of resource utilization distribution. in Proceedings of IEEE 21st International Symposium on Modeling, Analysis Simulation of Computer and Telecommunication Systems (MASCOTS 2013), Aug 2013: 370 – 374
- [44] 汪浩, 黄明, 龙浩. 基于 G/G/1 – FCFS、M/G/1 – PS 和 M/G/ ∞ 排队网络的 Web 服务组合性能分析. 计算机学报, Jan. 2013, vol. 36, no. 1: 22 – 38
- [45] D. Nicol, W. Sanders, and K. Trivedi. Model – based evaluation: From dependability to security. IEEE Transactions on Dependable and Secure Computing, Jan. Mar. 2004, vol. 1, no. 1: 48 – 65
- [46] K. Trivedi. Probability and Statistics with Reliability, Queuing, and Computer Science Applications. New York: John Wiley and Sons, 2001
- [47] B. Jansen, H. Ramasamy, M. Schunter, and A. Tanner. Architecting dependable and secure systems using virtualization. in Architecting Dependable Systems V, ser. Lecture Notes in Computer Science, R. Lemos, F. Giandomenico, C. Gacek, H. Muccini, and M. Vieira, Eds. Springer Berlin Heidelberg, 2008, vol. 5135: 124 – 149
- [48] J. B. Dugan and M. R. Lyu. Dependability Modeling for Fault – Tolerant Software and Systems. John Wiley and Sons Ltd. , 1995
- [49] B. Schneier. Secrets and Lies: Digital Security in a Networked World. John Wiley and Sons Ltd. , 2000
- [50] A. Iosup, S. Ostermann, M. Yigitbasi, R. Prodan, T. Fahringer, and D. H. J. Epema. Performance analysis of cloud computing services for many – tasks scientific computing. IEEE Transactions on Parallel and Distributed Systems, Jun. 2011, vol. 22, no. 6: 931 – 945
- [51] O. Tickoo, R. Iyer, R. Illikkal, and D. Newell. Modeling virtual machine performance: Challenges and ap-

- proaches. SIGMETRICS Performance Evaluation Review, Jan. 2010, vol. 37, no. 3: 55 – 60
- [52] Z. Zheng, Y. Zhang, and M. Lyu. Distributed QoS evaluation for real – world web services. in Proceeding of the 8th IEEE International Conference on Web Services (ICWS 2010), 2010: 83 – 90
- [53] Z. Zheng and M. Lyu. Collaborative reliability prediction of service – oriented systems. in Proceeding of ACM/IEEE 32nd International Conference on Software Engineering (ICSE 2010), vol. 1, May 2010: 35 – 44
- [54] Z. Zheng, H. Ma, M. Lyu, and I. King. Collaborative web service qos prediction via neighborhood integrated matrix factorization. IEEE Transactions on Services Computing, Jul. 2013, vol. 6, no. 3: 289 – 299
- [55] W. Qiu, Z. Zheng, X. Wang, X. Yang, and M. Lyu. Reputation – aware qos value prediction of web services. in Proceeding of the 10th IEEE International Conference on Services Computing (SCC 2013), Jun. 2013: 41 – 48
- [56] L. L. Peterson and B. S. Davie. Computer Networks: A Systems Approach. Elsevier, 2007
- [57] A. Heddaya and A. Helal. Reliability, availability, dependability and performability: A user – centered view. Boston University, Boston, MA, USA, Tech. Rep., 1997
- [58] X. Kong, J. Huang, C. Lin, and P. Ungsunan. Performance, fault – tolerance and scalability analysis of virtual infrastructure management system. in Proceeding of 2009 IEEE International Symposium on Parallel and Distributed Processing with Applications (ISPA 2009), Aug. 2009: 282 – 289
- [59] B. Schroeder and G. Gibson. A large – scale study of failures in high – performance computing systems. IEEE Transactions on Dependable and Secure Computing, Oct. Dec. 2010, vol. 7, no. 4: 337 – 350
- [60] C. Gunaratne, K. Christensen, B. Nordman, and S. Suen. Reducing the energy consumption of ethernet with adaptive link rate (alr). IEEE Transactions on Computers, Apr. 2008, vol. 57, no. 4: 448 – 461
- [61] K. Kritikos, B. Pernici, P. Plebani, C. Cappiello, M. Comuzzi, S. Benrernou, I. Brandic, A. Kertész, M. Parkin, and M. Carro. A survey on service quality description. ACM Computing Survey, Jul. 2013, vol. 46, no. 1: 1: 1 – 1: 58
- [62] M. Alrifai, D. Skoutas, and T. Risse. Selecting skyline services for qos – based web service composition. in Proceedings of the 19th International Conference on World Wide Web (WWW 2010), 2010: 11 – 20
- [63] L. Qi, Y. Tang, W. Dou, and J. Chen. Combining local optimization and enumeration for qos – aware web service composition. in Proceedings of the 8th IEEE International Conference on Web Services (ICWS 2010), Jul. 2010: 34 – 41
- [64] D. Skoutas, D. Sacharidis, A. Simitsis, and T. Sellis. Ranking and clustering web services using multicriteria dominance relationships. IEEE Transactions on Services Computing, Jul. 2010, vol. 3, no. 3: 163 – 177
- [65] Y. Y. Haimes, L. S. Lasdon, and D. A. Wismer. On a bicriterion formulation of the problems of integrated system identification and system optimization. IEEE Transactions on Systems, Man and Cybernetics, Jul. 1971, vol. 1, no. 3: 296 – 297
- [66] J. Tordsson, R. S. Montero, R. Moreno – Vozmediano, and I. M. Llorente. Cloud brokering mechanisms for optimized placement of virtual machines across multiple providers. Future Generation Computer Systems, Feb. 2012, vol. 28, no. 2: 358 – 367
- [67] D. Breitgand, A. Maraschini, and J. Tordsson. Policy – driven service placement optimization in federated clouds. IBM Research Division, Tech. Rep., Feb. 2011
- [68] R. Marler and J. Arora. Survey of multi – objective optimization methods for engineering. Structural and Multidisciplinary Optimization, 2004, vol. 26, no. 6: 369 – 395
- [69] K. Bessai, S. Youcef, A. Oulamara, C. Godart, and S. Nurcan. Bi – criteria workflow tasks allocation and scheduling in cloud computing environments. in Proceedings of IEEE 5th International Conference on Cloud

- Computing (CLOUD 2012), Jun. 2012: 638 – 645
- [70] H. Ma, F. Bastani, I. – L. Yen, and H. Mei. Qos – driven service composition with reconfigurable services. *IEEE Transactions on Services Computing*, Jan. – Mar. 2013, vol. 6, no. 1: 20 – 34
 - [71] P. Garcia Lopez, A. Montresor, D. Epema, A. Datta, T. Higashino, A. Iamnitchi, M. Barcellos, P. Felber, and E. Riviere. Edge – centric computing: Vision and challenges. *ACM SIGCOMM Computer Communication Review*, Sep. 2015, vol. 45, no. 5: 37 – 42
 - [72] M. H. Veatch. Approximate dynamic programming for networks: Fluid models and constraint reduction. Gordon College, Tech. Rep., 2005
 - [73] J. Van Velthoven, B. Van Houdt, and C. Blondia. Simultaneous transient analysis of QBD Markov chains for all initial configurations using a level based recursion. in *Proceedings of the 4th International Conference on the Quantitative Evaluation of Systems (QEST 2007)*, 2007: 79 – 88
 - [74] E. Chlebus and J. Brazier. Nonstationary poisson modeling of web browsing session arrivals. *Information Processing Letters*, 2007, vol. 102, no. 5: 187 – 190
 - [75] R. Guerra, J. Leite, and G. Fohler. Attaining soft real – time constraint and energy – efficiency in web servers. in *Proceedings of the 2008 ACM Symposium on Applied Computing (SAC 2008)*, 2008: 2085 – 2089
 - [76] J. Yuan, Y. Zheng, C. Zhang, W. Xie, X. Xie, G. Sun, and Y. Huang, T – drive: Driving directions based on taxi trajectories. in *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems (GIS 2010)*, 2010: 99 – 108
 - [77] J. Yuan, Y. Zheng, X. Xie, and G. Sun. Driving with knowledge from the physical world. in *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2011)*, 2011: 316 – 324
 - [78] M. Calzarossa, R. Marie, and K. Trivedi. System performance with user behavior graphs. *Performance Evaluation*, 1990, vol. 11, no. 3: 155 – 164
 - [79] D. Wang and K. Trivedi. Modeling user – perceived service availability. in *Service Availability*, ser. *Lecture Notes in Computer Science*, M. Malek, E. Nett, and N. Suri, Eds. Springer Berlin/Heidelberg, 2005, vol. 3694: 107 – 122
 - [80] M. Zyla and D. Caban. Dependability analysis of soa systems. in *Proceedings of the 3rd International Conference on Dependability of Computer Systems (DepCos – RELCOMEX 2008)*, Jun. 2008: 301 – 306
 - [81] Y. S. Dai, M. Xie, K. L. Poh, and G. Q. Liu. A study of service reliability and availability for distributed systems. *Reliability Engineering and System Safety*, 2003, vol. 79, no. 1: 103 – 112
 - [82] D. Oppenheimer and D. Patterson. Architecture and dependability of large – scale internet services. *IEEE Internet Computing*, Sep. 2002, vol. 6, no. 5: 41 – 49
 - [83] E. Jonsson. Towards an integrated conceptual model of security and dependability. in *Proceedings of the 1st International Conference on Availability, Reliability and Security (ARES 2006)*, Apr. 2006: 646 – 653
 - [84] I. Koren and C. M. Krishna. *Fault – Tolerant Systems*. San Francisco: Morgan Kaufmann, 2010
 - [85] Los Alamos National Laboratory. (2011) All systems failure/interrupt data 1996 – 2005. [Online]. Available: <http://institute.lanl.gov/data/fdata/>
 - [86] W. M. P. Van Der Aalst, A. H. M. Ter Hofstede, B. Kiepuszewski, and A. P. Barros. Workflow patterns. *Distrib. Parallel Databases*, Jul. 2003, vol. 14, no. 1: 5 – 51
 - [87] M. Malhotra and K. Trivedi. Dependability modeling using petri – nets. *IEEE Transactions on Reliability*, Sep. 1995, vol. 44, no. 3: 428 – 440
 - [88] J. Blake, A. Reibman, and K. Trivedi. Sensitivity analysis of reliability and performability measures for multiprocessor systems. in *Proceedings of the 1988 ACM SIGMETRICS Conference on Measurement and Modeling*

- of Computer Systems (SIGMETRICS 1988), 1988: 177 – 186
- [89] I. Toma, D. Roman, D. Fensel, B. Sapkota, and J. Gomez. A multi – criteria service ranking approach based on non – functional properties rules evaluation. in Proceedings of the 5th International Conference on Service – Oriented Computing (ICSOC 2007), 2007: 435 – 441
- [90] A. Ammar and D. Shah. Ranking: Compare, don't score. in Proceedings of the 49th Annual Allerton Conference on Communication, Control, and Computing (Allerton 2011), 2011: 776 – 783
- [91] Z. Zheng, X. Wu, Y. Zhang, M. Lyu, and J. Wang. QoS ranking prediction for cloud services. IEEE Transactions on Parallel and Distributed Systems, 2013, vol. 24, no. 6: 1213 – 1222
- [92] R. A. Bradley and M. E. Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. Biometrika, 1952, vol. 39, no. 3/4: 324 – 345
- [93] R. D. Luce. Individual Choice Behavior: A Theoretical Analysis. New York: Wiley, 1959
- [94] S. Negahban, S. Oh, and D. Shah. Iterative ranking from pair – wise comparisons. in Proceedings of the 26th Annual Conference on Neural Information Processing Systems (NIPS 2012), 2012: 2474 – 2482
- [95] I. C. F. Ipsen. Computing an eigenvector with inverse iteration. SIAM Review, 1997, vol. 39, no. 2: 254 – 291
- [96] I. S. Dhillon, B. N. Parlett, and C. Vömel. The design and implementation of the MRRR algorithm. ACM Transactions on Mathematical Software, Dec. 2006, vol. 32, no. 4: 533 – 560
- [97] W. J. Stewart. Numerical Solution of Markov Chains. CRC Press, 1991, vol. 8
- [98] E. Seneta. Sensitivity of finite Markov chains under perturbation. Statistics & probability letters, 1993, vol. 17, no. 2: 163 – 168
- [99] Y. Zhang, Z. Zheng, and M. Lyu. Exploring latent features for memory – based QoS prediction in cloud computing. in Proceedings of the 30th IEEE Symposium on Reliable Distributed Systems (SRDS 2011), 2011: 1 – 10
- [100] D. Schuller, U. Lampe, J. Eckert, R. Steinmetz, and S. Schulte. Cost – driven optimization of complex service – based workflows for stochastic QoS parameters. in Proceedings of IEEE 19th International Conference on Web Services (ICWS 2012), Jun. 2012: 66 – 73
- [101] S. Liu, G. Quan, and S. Ren. On – line real – time service allocation and scheduling for distributed data centers. in Proceedings of the 8th IEEE International Conference on Services Computing (SCC 2011), Jul. 2011: 528 – 535
- [102] L. Minas and B. Ellison. Energy efficiency for information technology: How to reduce power consumption in servers and data centers. Intel Press, 2009
- [103] V. Pallipadi. Enhanced intel speedstep technology and demand – based switching on Linux. Intel Developer Service, 2008
- [104] Intel turbo boost technology – on – demand processor performance. [Online]. Available: <http://www.intel.com/content/www/us/en/architecture-and-technology/turbo-boost/turbo-boost-technology>
- [105] AMD PowerNow! Technology. [Online]. Available: <http://www.amd.com/us/products/technologies/amd-powernow-technology/Pages/amd-powernow-technology.aspx>
- [106] AMD Cool'n'Quiet Technology. [Online]. Available: <http://www.amd.com/us/products/technologies/cool-n-quiet/Pages/cool-n-quiet.aspx>
- [107] L. Chen and N. Gautam. Server frequency control using Markov decision processes. in Proceedings of IEEE INFOCOM 2009, Apr. 2009: 2951 – 2955
- [108] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. Numerical Recipes 3rd Edition: The Art of Scientific Computing, 3rd ed. New York: Cambridge University Press, 2007

- [109] C. D. Meyer, Ed. . Matrix Analysis and Applied Linear Algebra. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2000
- [110] C. D. Meyer and J. M. Shoaf. Updating finite Markov chains by using techniques of group matrix inversion. *Journal of Statistical Computation and Simulation*, 1980, vol. 11, no. 3 – 4: 163 – 181
- [111] A. Gandhi, V. Gupta, M. Harchol – Balter, and M. Kozuch. Optimality analysis of energy – performance trade – off for server farm management. *Performance Evaluation*, 2010, vol. 67, no. 11: 1155 – 1171
- [112] M. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, 1994
- [113] M. Wooldridge. *An Introduction to Multiagent Systems*. John Wiley & Sons, 2009
- [114] E. M. Maximilien and M. P. Singh. Multiagent system for dynamic web services selection. in *Proceeding of first Workshop on Service – Oriented Computing and Agent – Based Engineering (SOCABE at AA – MAS)*, 2005: 25 – 29
- [115] E. Val, M. Rebollo, and V. Botti. Enhancing decentralized service discovery in open service – oriented multi – agent systems. *Autonomous Agents and Multi – Agent Systems*, Oct. 2012: 1 – 30
- [116] K. M. Sim. Agent – based cloud computing. *IEEE Transactions on Services Computing*, 2012, vol. 5, no. 4: 564 – 577
- [117] E. Al – Masri and Q. H. Mahmoud. Investigating web services on the world wide web. in *Proceeding of the 17th International Conference on World Wide Web 2008 (WWW 2008)*, 2008: 795 – 804
- [118] D. Zhu, R. Melhem, and D. Mosse. The effects of energy management on reliability in real – time embedded systems. in *Proceedings of IEEE/ACM International Conference on Computer Aided Design (ICCAD 2004)*, Nov. 2004: 35 – 40
- [119] X. Qi, D. Zhu, and H. Aydin. Global reliability – aware power management for multiprocessor real – time systems. in *Proceedings of IEEE 16th International Conference on Embedded and Real – Time Computing Systems and Applications (RTCSA 2010)*, Aug. 2010: 183 – 192
- [120] J. M. Reason and J. M. Rabaey. A study of energy consumption and reliability in a multi – hop sensor network. *ACM SIGMOBILE Mobile Computing and Communications Review*, Jan. 2004, vol. 8, no. 1: 84 – 97
- [121] M. Lanus, L. Yin, and K. Trivedi. Hierarchical composition and aggregation of state – based availability and performability models. *IEEE Transactions on Reliability*, Mar. 2003, vol. 52, no. 1: 44 – 52
- [122] C. Derman. Denumerable state markovian decision processes – average cost criterion. *The Annals of Mathematical Statistics*, Dec. 1966, vol. 37, no. 6: 1545 – 1553

反侵权盗版声明

电子工业出版社依法对本作品享有专有出版权。任何未经权利人书面许可，复制、销售或通过信息网络传播本作品的行为；歪曲、篡改、剽窃本作品的行为，均违反《中华人民共和国著作权法》，其行为人应承担相应的民事责任和行政责任，构成犯罪的，将被依法追究刑事责任。

为了维护市场秩序，保护权利人的合法权益，本社将依法查处和打击侵权盗版的单位和个人。欢迎社会各界人士积极举报侵权盗版行为，本社将奖励举报有功人员，并保证举报人的信息不被泄露。

举报电话：(010) 88254396；(010) 88258888

传 真：(010) 88254397

E-mail: dbqq@phei.com.cn

通信地址：北京市海淀区万寿路 173 信箱

电子工业出版社总编办公室

邮 编：100036